



# تمارين لبرمجة الحاسوب

للطالب المبدع  
ايهم حاتم

إرادة - ثقة - تغيير

# Practice Solutions

---

- prepared by: Iham Abu-Shabib.
  - For contact via email: [ihamabushabib@gmail.com](mailto:ihamabushabib@gmail.com).
- Note: Chapters that are not mentioned doesn't have any practices.

## CHAPTER 2: BASIC ELEMENTS OF C++

---

- PRACTICE p.23: Knowing that 1 foot has 12 inches, write a program to calculate number of feet and the remaining inches in 500 inches

### INPUT

```
const int inch_per_foot=12;
int totalInches=500,totalFeet,remainingInches;

totalFeet=totalInches/inch_per_foot;
remainingInches=totalInches%inch_per_foot;

cout <<"total Feets="<<totalFeet<<endl;
cout <<"remaining Inches="<<remainingInches<<endl;
```

### OUTPUT

```
total Feets=41
remaining Inches=8
```

- PRACTICE p28.1: What is the output of the following code snippet?

### INPUT

```
// Remember: a boolean variable can only store values 0 or 1
bool b1 = 10 * 6 - 60;      // b1 = 0
bool b2 = 3.14;           // b2 = 1
bool b3 = 0.04;           // b3 = 1
bool b4 = 'a';            // b4 = 1
cout<<b1<<"\t"<<b2<<"\t"<<b3<<"\t"<<b4;
```

### OUTPUT

```
0      1      1      1
```

- PRACTICE p28.2: What is the output of the following code snippet?

## INPUT

```
bool b = 10;           // b = 1
int v1 = 13.7;         // v1 = 13
int v2 = 0.999;       // v2 = 0
int v3 = 'a';         // v3 = 97
int v4 = b;           // v4 = 1
int v5 = false;       // v5 = 0
cout<<v1<<"\t"<<v2<<"\t"<<v3<<"\t"<<v4;
```

## OUTPUT

```
13    0    97    1
```

- Note: After we enter 0.9-17 times, the value of v1 will be rounded to one

## INPUT

```
int v2 = 0.999999999999999999;
cout<<v2;
```

## OUTPUT

```
1
```

- PRACTICE p28.3: What is the output of the following code snippet?

## INPUT

```
bool b = 0;
char c1 = 'A';           // c1 = A
char c2 = 66;           // 66 is the ASCII code of 'B' => c1 = B
char c3 = 97.65;        // 97 is the ASCII code of 'a' => c2 = a , 97 because char
is integral data type
char c4 = b;           // 0 is the ASCII code of null => c3 store null
cout<<c1<<c2<<c4<<c3;
```

## OUTPUT

```
ABa//note that null is an empty character that will not be printed(not a space)
```

- PRACTICE P28.4: What is the output of the following code snippet?

## INPUT

```
bool b = 10;           // b = 1
double d1 = 4;        // d1 = 4.0
double d2 = b / 2.0;  // d2 = 0.5
double d3 = 'B';     // d3 = 66.0
double d4 = 'a';     // d4 = 97.0
cout<<d1<<d2<<d3<<d4;
```

## OUTPUT

```
40.56697
```

## CHAPTER 4: CONTROL STRUCTURES I (SELECTION)

- PRACTICE p5: What is the output if num = 13?

### INPUT

```
cout<< (0 <= num <= 10); //0<=13 => 1
                          //1<=10 => 1
```

### OUTPUT

```
1
```

- PRACTICE p9: Solve examples 4-8 and 4-9 using the or ( || ) and not ( ! ) operators.
- example 4-8

## SOLUTION

```
!((num<20) || (num>100))
```

- example 4-9
- **SOLUTION**

```
!((ch < 'A') || (ch > 'Z'))
```

- PRACTICE P.16: Write a program to compute and output the sales tax and the final price of an item sold in a particular state. Knowing that, the sales tax is calculated as follows: The state's portion of the sales tax is 4%, and the city's portion of the sales tax is 1.5%. If the item is a luxury item (items with price over \$10,000), then there is a 10% luxury tax. The final price of

the item is the selling price in addition to the sales tax.

## SOLUTION

```
double sellingPrice, finalPrice, salesTax;
const double stateTax=0.04, cityTax=0.015, luxuryTax=0.1;
// طبعا ممكن ما نعملهم ثوابت عشان نوفر ميموري

cout <<"please enter your selling price.\n";
cin>>sellingPrice;

salesTax=(stateTax+cityTax)*sellingPrice;

if(sellingPrice>=10000)
    salesTax=(stateTax+cityTax+luxuryTax)*sellingPrice;

finalPrice=sellingPrice+salesTax;

cout <<"The sales tax="<<salesTax<<endl;
cout <<"price after tax="<<finalPrice<<endl;
```

- PRACTICE p17: What is the output if user input is -5?

## INPUT

```
int x;
cout << "Enter a positive integer value..."<<endl;
cin >> x; //x=-5
if (0<= x <= 10) //0<=-5 =>0 //0<=10 => 1
    cout << x <<" is less than 10";
else
    cout << x <<" is greater than 10";
```

## OUTPUT

```
-5 is less than 10
```

- PRACTICE p23: Write the previous program using one-way selection.

### example 4-35

```
if (gender == 'M')
    if (age < 21 )
        policyRate = 0.05;
    else
        policyRate = 0.035;
else if (gender == 'F')
    if (age < 21 )
        policyRate = 0.04;
    else
        policyRate = 0.03;
```

### SOLUTION

```
if (gender == 'M' && age < 21 )
    policyRate = 0.05;
if (gender == 'M' && age >= 21 )
    policyRate = 0.035;
if (gender == 'F' && age < 21 )
    policyRate = 0.04;
if (gender == 'F' && age >= 21 )
    policyRate = 0.03;
```

- PRACTICE p.25: Write a program to calculate the monthly paycheck of a salesperson at a local department store. Knowing that every salesperson has a base salary. The salesperson also receives a bonus at the end of each month, based on the following criteria: If the salesperson has been with the store for five years or less, the bonus is \$10 for each year that he or she has worked there. If the salesperson has been with the store for more than five years, the bonus is \$20 for each year that he or she has worked there. The salesperson can earn an additional bonus as follows: If the total sales made by the salesperson for the month are at least \$5,000 but less than \$10,000, he or she receives a 3% commission on the sale. If the total sales made by the salesperson for the month are at least \$10,000, he or she receives a 6% commission on the sale.

### SOLUTION

```
float BaseSalary, Bonus, AdditionalBonus, Monthly_Paycheck, TotalSales;
int YearsWorked;

cout <<"please enter your years of work ,your base salary,your Total sales"
<<endl;
cin >>YearsWorked>>BaseSalary>>TotalSales;

if (YearsWorked<=5)
    Bonus=10*YearsWorked;
else
    Bonus=20*YearsWorked;
```

```

if (TotalSales<5000)
    AdditionalBonus=0;

else if (TotalSales>=5000 && TotalSales<10000)
    AdditionalBonus=0.03*TotalSales;

else if (TotalSales>=10000)
    AdditionalBonus=0.06*TotalSales;

Monthly_Paycheck=BaseSalary+Bonus+AdditionalBonus;

cout <<"your paycheck of this month:"<<Monthly_Paycheck<<endl;

```

- PRACTICE p.32: What is the output if the user input is 2, 3, 4, 6, and 7?

## input

```

int num;
cout << "Enter an integer between 0 and 5: ";
cin >> num;

switch(num)
{
case 0:
case 1:
    cout << "One";
case 2:
    cout << "Two";
case 3:
    cout << "Three" << endl;
    break;
case 4:
    break;
case 5:
    cout << "Five";
default:
    cout << "Invalid Input" << endl;
}

```

## OUTPUT

```

//2 =>TwoThree
//3 =>Three
//4=> (فراغ)
//6=>Invalid Input

```

- PRACTICE p.33: Rewrite the implementation of example 4-37 using switch statement:

## example 4-37

- Given a student test score (out of 100), write a program to calculate his/her grade (A, B, C, D, F). The grade is assigned as follows: If the average test score is greater than or equal to 90, the grade is A; if the average test score is greater than or equal to 80 and less than 90, the grade is B; if the average test score is greater than or equal to 70 and less than 80, the grade is C; if the average test score is greater than or equal to 60 and less than 70, the grade is D; otherwise, the grade is F.

// محلول بالاسلايدات ولكن سيتم وضعه هنا لغايات التجميع

```
int score;
char grade;
cout << "Enter your test score... " << endl;
cin >> score;
if (score >= 90)
    grade = 'A';
else if (score >= 80)
    grade = 'B';
else if (score >= 70)
    grade = 'C';
else if (score >= 60)
    grade = 'D';
else
    grade = 'F';
cout << "student grade is: " << grade;
```

## SOLUTION

```
int score;
char grade;
cout << "Enter your score..." << endl;
cin >> score;
switch (score / 10)
{
case 0:
case 1:
case 2:
case 3:
case 4:
case 5:
    grade = 'F';
    break;
case 6:
    grade = 'D';
    break;
case 7:
    grade = 'C';
    break;
case 8:
    grade = 'B';
    break;
```



```

case 9:
case 10:
    grade = 'A';
    break;
default:
    cout << "Invalid test score." << endl;
}
cout<<"Student Grade is: "<<grade;

```

## CHAPTER 5:CONTROL STRUCTURES II (REPETITION)

- PRACTICE P11: On a standard telephone keypad, the letters A-Z are mapped onto the phone number digits 0-9 as shown in the following diagram. A,B,C are mapped to 2, and D,E,F are mapped to 3, and so on. Write a program to read the letter codes A to Z from user and prints the corresponding telephone digit. Your program should continue read letters until # is read. You may ignore invalid inputs.

### SOLUTION

```

char sentinel = '#';
char letter;

cout << "To stop the program enter #." << endl;

cout << "please enter a letter\n";
cin >> letter;

while (letter != sentinel)
{
    switch (letter)
    {

        case ('A'):
        case ('B'):
        case ('C'):
            cout << "2" << endl;
            break;
        case ('D'):
        case ('E'):
        case ('F'):
            cout << "3" << endl;
            break;

        case ('G'):
        case ('H'):
        case ('I'):
            cout << "4" << endl;
            break;

```

```

case ('J'):
case('K'):
case ('L'):
    cout <<5<<endl;
    break;

case ('M'):
case('N'):
case ('O'):
    cout <<6<<endl;
    break;

case ('P'):
case('Q'):
case ('R'):
case ('S'):
    cout <<7<<endl;
    break;

case ('T'):
case('U'):
case ('V'):
    cout <<8<<endl;
    break;

case ('W'):
case('X'):
case ('Y'):
case ('Z'):
    cout <<9<<endl;
    break;

default:
    cout <<"wrong input";

}

cout <<"please enter a letter\n";
cin>>letter;
}

```

- PRACTICE p15: On a standard telephone keypad, the letters A-Z are mapped onto the phone number digits 0-9 as shown in the following diagram. A,B,C are mapped to 2, and D,E,F are mapped to 3, and so on. Write a program to read the letter codes A to Z from user and prints the corresponding telephone digit. Your program should continue read letters until it reads any non-letter inputs.

## SOLUTION

```
bool flag =false;
char letter;

while (!flag)
{
    cout <<"please enter a letter\n";
    cin>>letter;

    switch (letter)
    {

        case ('A'):
        case('B'):
        case ('C'):
            cout <<2<<endl;
            break;
        case ('D'):
        case('E'):
        case ('F'):
            cout <<3<<endl;
            break;

        case ('G'):
        case('H'):
        case ('I'):
            cout <<4<<endl;
            break;

        case ('J'):
        case('K'):
        case ('L'):
            cout <<5<<endl;
            break;

        case ('M'):
        case('N'):
        case ('O'):
            cout <<6<<endl;
            break;

        case ('P'):
        case('Q'):
        case ('R'):
        case ('S'):
            cout <<7<<endl;
            break;

        case ('T'):
        case('U'):
        case ('V'):
            cout <<8<<endl;
            break;
```

```

    case ('w'):
    case('x'):
    case ('Y'):
    case ('z'):
        cout <<9<<endl;
        break;

    default:
        cout <<"wrong input";
        flag=true;

}

}

```

- PRACTICE p.22: Write a program to read 10 integers and then prints the maximum number among them.

## SOLUTION

```

int max=0;
// قيمة مناسبة في حال دخل المستخدم سالب رح تظل اكبر قيمة وفي حال موجب رح تتعدل
// مع الشرط طريقة اخرى للحل هي المثال اللي تحت
int number;

for (int i=1;i<=10;i++)
{
    cout<<"enter your value.\n";
    cin >>number;

    if (max<number)
        max=number;
}

cout <<"The maximum number among your entries is:"<<max<<endl;

```

- additional practice: Write a program to read 10 integers and then prints the minimum number among them.

## SOLUTION

```

int min;
int number;

cout<<"enter your value.\n";

```

```

cin >>number;
min=number;

for (int i=2;i<=10;i++) //عشان اعتبرنا اول مدخل رقم واحد
{

    if (number < min)//للتحديثات الجديدة على الرقم
        min=number;

    cout<<"enter your value.\n";
    cin >>number;

}

cout <<"The minimum number among your entries is:"<<min<<endl;

```

- PRACTICE p.25: It is known that an integer  $n$  is divisible by 3 if and only if the sum of its digits is divisible by 3. Write a program to check if a number entered from user is divisible by 3 or not, then print the proper message

## SOLUTION

```

int num,sum=0;

cout <<"please enter the number\n";
cin >>num;

do
{
    sum+=num%10;
    num/=10;

}
while (num!=0);

cout <<"the sum of the digits is ="<<sum<<endl;

if (sum%3==0)
    cout <<"the number is divisible by three.\n";
else
    cout <<"the number is not divisible by three.\n";
// طبعا ما بنحتاج نعمل جمع للحدود نقدر مباشرة نستخدم باقي القسمة ولكن للتدريب
على المثال

```

PRACTICE p.27: Rewrite the previous example using flag-controlled while loop.

EXAMPLE 5-40: Write a program that reads integer numbers continuously from user and calculate the sum of positive numbers only. Your program should stop reading once an input failure occurred or the user entered a negative number.

## SOLUTION

```
int num, sum=0;
bool flag=false;

while (!flag)
{
    cout <<"enter a number.\n";
    cin>>num;

    if (!cin || num<0)
        flag=true;

    sum+=num;
}
cout <<"the sum is ="<<sum<<endl;
```

PRACTICE p30: Write a program to create the the following pattern.

```
*****
****
***
**
*
```

- Modify your program so that you read number of rows from user.

## SOLUTION

```
int rows;
cout <<"please enter number of rows.\n";
cin >>rows; //5

for (int i=1;i<=rows;i++) //عشان اللوب الخارجي يتحكم بعدد الصفوف والداخلي
الاعمدة
{

    for (int j=rows;j>=i;j--)
        cout <<"*";

    cout <<endl;
}
```

## CHAPTER 6: USER-DEFINED FUNCTIONS

- PRACTICE p.9: Write the definition of function absolute that takes a decimal number as a parameter, and returns its absolute value. Test the correctness of your function.

### SOLUTION

```
double absolute(double num)
{
    if (num>=0)
        return num;

    return -num;
}
```

- PRACTICE p11.1: Modify this program so that it allows the user to enter the desired sum of the numbers to be rolled.
  - EXAMPLE 6-11: Write a function that rolls a pair of dice until the sum of the numbers rolled is a specific number, and return the number of times the dice are rolled to get the desired sum. Test the correctness of your function.

```
int rollDice(int num)
{
    int die1, die2, sum;
    srand(time(0));
    int rollCount = 0;
    if(num >= 2 && num <= 12)
    {
        do
        {
            die1 = rand() % 6 + 1;
            die2 = rand() % 6 + 1;
            sum = die1 + die2;
            rollCount++;
        }
        while (sum != num);
    }
    return rollCount;
}

int main()
{
    cout<< "The number of times the dice are rolled to "
    << "get the sum 15 = " << rollDice(15) << endl;
    cout<< "The number of times the dice are rolled to "
    << "get the sum 10 = " << rollDice(10) << endl;
}
```

## SOLUTION

```
int rollDice()
{

    int num;
    cout <<"please enter the desired sum of the numbers to be rolled."<<endl;
    cin>>num;

    int die1, die2, sum;
    srand(time(0));
    int rollCount = 0;

    if(num >= 2 && num <= 12)
    {
        do
        {
            die1 = rand() % 6 + 1;
            die2 = rand() % 6 + 1;
            sum = die1 + die2;
            rollCount++;
        }
        while (sum != num);
    }
    return rollCount;
}

int main()
{
    cout <<rollDice()<<endl;
}
```

- PRACTICE p11.2: A string is a palindrome if it reads forward and backward in the same way. For example, the strings "madam", "5", "434", and "789656987" are all palindromes. write the definition of a C++ function isPalindrome that returns true if a string is a palindrome and false otherwise.



```

bool isPalindrome(string str)
{
    int x=str.length()-1;//آخر اندكس فالسترينغ
    for (int i=0;i<str.length()/2;i++) // نمشي لنص السترينغ عشان ما نكرر العملية
        if (str[i]!=str[x-i])
            return 0;

    return 1;
}

```

- PRACTICE p12: Rewrite example 6-13 so that you place the functions prototype before the definition of function main.
  - EXAMPLE 6-13: Use function Larger we studied earlier to write the definition of function compareThree that determine the largest of three numbers.

## SOLUTION

```

double larger(double , double );
double compareThree(double , double , double );

int main()
{
    double one = 13.00;
    cout<< compareThree(one, 29,34) << endl;
}

double larger(double x, double y)
{
    if (x >= y)
        return x;
    return y;
}

double compareThree(double x, double y, double z)
{
    return larger(x, larger(y, z));
}

```

- PRACTICE p19: What is the output of the following program.

```

void addFirst(int& first, int& second);
void doubleFirst(int one, int two);
void squareFirst(int& ref, int val);
int main()
{
    int num = 5;

```

```

    cout << "main: num = " << num << endl;
    addFirst(num, num);
    cout << "main after addFirst: num = " << num << endl;
    doubleFirst(num, num);
    cout << "main after doubleFirst: num = " << num << endl;
    squareFirst(num, num);
    cout << "main after squareFirst: num = " << num << endl;
}

void addFirst(int& first, int& second)
{
    cout << "Inside addFirst:"<<endl;
    cout << "first = "<< first << ", second = " << second << endl;
    first = first + 2;
    second = second * 2;
    cout << "first = "<< first << ", second = " << second << endl;
}

void doubleFirst(int one, int two)
{
    cout << "Inside doubleFirst:"<<endl;
    cout<<"one = "<< one << ", two = " << two << endl;
    one = one * 2;
    two = two + 2;
    cout<<"one = "<< one << ", two = " << two << endl;
}

void squareFirst(int& ref, int val)
{
    cout << "Inside squareFirst: "<<endl;
    cout <<"ref = "<< ref << ", val = " << val << endl;
    ref = ref * ref;
    val = val + 2;
    cout <<"ref = "<< ref << ", val = " << val << endl;
}

```

## SOLUTION

```

main: num = 5
Inside addFirst:
first = 5, second = 5
first = 14, second = 14
main after addFirst: num = 14
Inside doubleFirst:
one = 14, two = 14
one = 28, two = 16
main after doubleFirst: num = 14
Inside squareFirst:
ref = 14, val = 14
ref = 196, val = 16
main after squareFirst: num = 196

```

- PRACTICE p24: Use the scope resolution operator in the example given in the second rule to access global identifiers to call function one in function three.

## SOLUTION

```
void one(int x, char y);
void three(int one, double y, int z);

int main()
{
    int num = 7, first = 8;
    double x = 13.5;
    char last = 'd';
    cout<<"main : "<<num<<"\t"<<first<<"\t"
        <<x<<"\t"<<last<<endl;
    one(first, last);
    three(num,x,first);
}
void one(int x, char y)
{
    cout<<"one : "<<x<<"\t"<<y<<endl;
}
void three(int one, double y, int z)
{
    ::one(3, '2');
    cout<<"three: "<<one<<"\t"<<y<<"\t"<<z<<endl;
}
```

## CHAPTER 8: ARRAYS AND STRINGS

- PRACTICE p.5: Write the required C++ program to check if two arrays are equal (has the same elements in the same order).

## SOLUTION

```
const int SIZE=5;

int list1[SIZE]={1,2,3,4,5};
int list2[SIZE]={1,2,3,4,5};

int i;
for ( i=0;i<SIZE;i++)
    if (list1[i]!=list2[i])
    {
        cout <<"not equal\n";
        break;
    }

if (i==SIZE)//ليست جزء من اللوب
```

```
cout <<"equal\n";
```

## another solution

```
bool flag=false;
int counter=0;
while(!flag &&counter<SIZE)
{
    if (list1[counter]!=list2[counter])
        flag=true;

        counter++;
}

if (flag==true)
    cout <<"not equal\n";
else
    cout <<"equal\n";
```

- PRACTICE p.6: Write the required code to find the index of the last occurrence of the smallest element in the array **sales**.

## SOLUTION

```
const int SIZE=5;

int sales[SIZE]= {1,0,3,2,0};

int min=sales[0];
int minIndex=0;

for (int i=1; i<SIZE; i++)
    if (min >= sales[i])
    {
        min=sales[i];
        minIndex=i;
    }

cout <<"the minimum element is:"<<min<<" and it\'s last position is:"
<<minIndex;
```

- PRACTICE p8: Rewrite the sequential search program using loop and break statement.

## SOLUTION

```
const int SIZE=5;

int sales[SIZE];

srand(time(0));

for (int i=0; i<SIZE; i++)
    sales[i]=rand()%10+1;

for (int i=0; i<SIZE; i++)
    cout <<setw(5)<<sales[i];

cout <<endl;

int num;
cout <<"enter the num you want to find\n";
cin>>num;

bool found=false;
int index=0;

for (int i=0; i<SIZE; i++)
{
    if (num==sales[i])
    {
        found=true;
        index=i;
        break;
    }
}

if (found)
    cout <<"found at "<<index<<endl;
else
    cout <<"not found"<<endl;
```

- PRACTICE p13: Write the required C++ program to copy a C-string into another C-string variable.

## SOLUTION

```
char text1[]="ahmad";
char text2[]="ama1";

for (int i=0; i<5; i++) //5 is size of the first array
{

    if (text2[i]=='\0')
    {
        text1[i]='\0';
        break;
    }
}
```

```

    }

    text1[i]=text2[i];

}

// اظهار النتيجة
cout <<"tex1 =";

for (int i=0; i<5; i++)
{

    if (text1[i]=='\0')
        break;

    cout <<text1[i];

}

// cout بنقدر نستخدم جملة
// مباشرة ورح تطلع نفس النتيجة بدل اللوب
//cout <<text1;

```

- PRACTICE p15: What is the output of the following code snippet?

```

char name[10] = "Ahmad Ali";
name[4] = 0;
cout << name;

```

## OUTPUT

Ahma

- PRACTICE p24: Write the required C++ code to find the sum of each column separately and store them in an array.

## SOLUTION

```

const int NUMBER_OF_ROWS = 4;
const int NUMBER_OF_COLUMNS = 5;
int matrix[NUMBER_OF_ROWS][NUMBER_OF_COLUMNS];

srand(time(0));

for (int r=0;r<NUMBER_OF_ROWS;r++)
    for (int c=0;c<NUMBER_OF_COLUMNS;c++)
        matrix[r][c]=rand()%10;

int sum[NUMBER_OF_COLUMNS]={0};
int AllSum=0;

```

```

for (int c = 0; c < NUMBER_OF_COLUMNS; c++)
{
for ( int r = 0; r < NUMBER_OF_ROWS; r++)
sum[c] = sum[c] + matrix[r][c];

cout <<"sum of coulums "<<c+1<<" ="<<sum[c]<<endl;

AllSum+=sum[c];
}

cout <<"sum of all elements ="<<AllSum<<endl;

```

- PRACTICE p24: Write the required C++ program to determine the largest element in each row and print them.

## SOLUTION

```

const int NUMBER_OF_ROWS = 4;
const int NUMBER_OF_COLUMNS = 5;
int matrix[NUMBER_OF_ROWS][NUMBER_OF_COLUMNS];

srand(time(0));

for (int r=0;r<NUMBER_OF_ROWS;r++)
    for (int c=0;c<NUMBER_OF_COLUMNS;c++)
        matrix[r][c]=rand()%10;

for (int r=0;r<NUMBER_OF_ROWS;r++)
{

    for (int c=0;c<NUMBER_OF_COLUMNS;c++)
        cout <<setw(5)<<matrix[r][c];
        cout <<endl;
}

// PRACTICE: Write the required C++ program to determine the largest element in
// each row and print them.

int largest[NUMBER_OF_COLUMNS]={0};

for (int r=0;r<NUMBER_OF_ROWS;r++)
{
    for (int c=0;c<NUMBER_OF_COLUMNS;c++)
        if (matrix[r][c]>largest[r])
            largest[r]=matrix[r][c];

    cout <<"the maximum number in row"<<r+1<<" ="<<largest[r]<<endl;
    // ممكن بداله نعمل لوب بس لظهار عناصره بس بما انهم
    // للصفوف parallel استخدمتها هان

```

```

}
// الحل باللوب بدل السطر
/*for (int i=0;i<NUMBER_OF_COLUMNS<i++)
    cout <<"the maximum number in row"<<r+1<<" ="<<largest[r]<<endl; */

```

- PRACTICE p30: Use function Larger we studied earlier to write the definition of function Largest that determine the largest number in the array.

## SOLUTION

```

int larger(int num1,int num2)
{
//    return (num1>num2? num1:num2);

if (num1>num2)
    return num1;

    return num2;

}

int largest(int mylist[], int SIZE)
{

int maximum=mylist[0];

for (int i=1;i<SIZE;i++)
    {

        if(maximum<larger( maximum , mylist[i]) )
            maximum=larger( mylist[i] , mylist[i] );

    }

    return maximum;

}

int main()
{
    srand(time(0));
    int mylist[5];

    for(int i=0;i<5;i++)
        mylist [i]=rand()%10+1;
}

```



```

for(int i=0;i<5;i++)
    cout<<setw(5)<<mylist [i];

    cout <<endl;

cout <<"the maximum number in the array is:"<<largest(mylist,5);
}

```

## PREVIOUS YEARS PRACTICE EXAM QUESTIONS

- Write the implementation of function **GCD** that takes as a parameter two integer number your function should check if the passed parameters are positive then calculate and print out the greatest common divisor **GCD** between them otherwise print a proper message.

(hint **GCD** is the largest positive integer that divides each of the integers )

For example :

calling the function **GCD** (8,12) will print out 4 .

calling the function **GCD** (8,15) will print out 1 .

calling the function **GCD** (4,16) will print out for 4.

calling the function **GCD** (-4,16) word print out invalid values.

## SOLUTION

```

void GCD(int num1, int num2)
{
    if (num1<=0 || num2<=0 )
    {

        cout <<"invalid input!\n";
        return; // بطلع من الفنكشن وما بكمل
    }
    int Minimum_Number=0; // بنحدد الرقم الاصغر عشان لو قسمنا الاصغر عالاكبر مستحيل
    الباقي صفر فينوفر عمليات

    if(num1<num2)
        Minimum_Number=num1;
    else
        Minimum_Number=num2;

    int Greatest_Common_Devisor=1;

    for(int i=1; i<=Minimum_Number; i++)
        if(num1 % i==0 && num2 % i==0)
            Greatest_Common_Devisor = i; // كل مرة بتجدد القاسم الاكبر كل ما بزيد
    العداد
}

```

```

        cout <<"The GCD of (" <<num1 << ","<<num2 <<") is:"
        <<Greatest_Common_Divisor<<endl;

    }

int main()
{
    GCD(8,12);
    GCD(8,15);
    GCD(4,16);
    GCD(-4,16);

}

```

- Write the definition of function **PrintStars** that takes two integer value as a parameters (represent the number of rows and columns). For example: if the input is 5 and 3 the function prints five rows and three columns of stars.
  - Please call the function **PrintStars** to bring the following:

(A) ***** ***** ***** *****	(B) ***** ***** *****
---	--------------------------------

## SOLUTION

```

void PrintStars(int rows,int columns)
{
    for (int i=1;i<=rows;i++)
    {
        for (int j=1;j<=columns;j++)
            cout<<"*";
        cout <<endl;
    }
}

int main()
{

```

```
cout <<"(A)"<<endl;
PrintStars(4,4);
cout <<"(B)"<<endl;
PrintStars(3,6);

}
```

- Write the definition of a function **PrintStars** that takes one integer value as a parameter (represents the number of rows).

For example: if the input is 5, the function prints 5 rows of stars "", starting with one star at row one and 5 stars at row 5.

- The output after the function call **PrintStars** (5) should be:

```
*
**
***
****
*****
```

- -The output after the function call **PrintStars** (3) should be:

```
*
**
***
```

- \*In the main function, Prompt the user to enter one integer value, then use the function **PrintStars** to print the pattern of the stars.

```
void PrintStars(int rows)
{
    for (int i=1; i<=rows; i++)
    {
        for (int j=1; j<=i; j++)
            cout<<"*";

        cout <<endl;
    }
}

int main()
{
    int num;
    cout <<"please enter an integer value\n";
    cin>>num;
    PrintStars(num);
}
```

- write of implementation of a function **RectangularPrismVol** that takes three decimal parameters , the three parameters represent the height ,width and length. Your function should calculate and return the volume of the rectangular prism it also should ensure that the passed parameters are positive numbers otherwise set the values to zero  
example **RectangularPrismVol(2,20,10)** should return 400.

Hint: volume =height \* width \* length.

- find the volume for **rectangular prism1** with width=10 ,height=5 and length=3 ,and for **rectangular prism2** with width=10 ,height=0.5 and length=4.

## SOLUTION

```
double RectangularPrismVol(double height,double width,double length)
{
    if (height<=0||width<=0||length<=0)// اذا غيرنا قيمة احدهم لصفر بدل سالب رح
    يرجع الضرب صفر كمان
        return 0;

    return height * width * length;
}

int main()
{
    cout <<"the volume for rectangular prism1 ="<<RectangularPrismVol(5,10,3)
<<endl;
    cout <<"the volume for rectangular prism2 ="<<RectangularPrismVol(0.5,10,4)
<<endl;
}
```

- Write the definition of a function that takes 4 decimal numbers as input. Those numbers represent the coordinates of two points on a line (x1,y1), (x2,y2), and return the line slope. Note: a slope of a line is calculated using the following formula slope = (y2-y1)/(x2-x1)  
Examples:
  - If the function input is (3, 7), and (7, 14), then the returned value is 1.75
  - If the function input is -1, 5, 4, -7, then the returned value is -2.4

## SOLUTION

```
double slope(double x1,double y1,double x2,double y2)
{
    return (y2-y1)/(x2-x1);
}

int main()
{
    cout <<"slope of (3, 7) and (7, 14):"<<Slope(3,7,7,14)<<endl;
    cout <<"slope of (-1, 5) and ( 4, -7):"<<Slope(-1, 5, 4, -7)<<endl;
}
```

- write the definition of a function that takes an integer number and as an input and perform the following equation:

$$f(n) = \begin{cases} \frac{n}{2} & \text{if } n \text{ is even} \\ 3n + 1 & \text{if } n \text{ is odd} \end{cases}$$

In the main function prompt the user to enter an integer number a, then use the function to print f(a) and f(a+1) and the difference between the two outputs.

## SOLUTION

```
double f(int n)
{
    if(n%2==0)
        return n/2.0;

    return 3*n+1;
}

int main()
{
    int a;
    cout <<"please enter the number a\n";
    cin>>a;

    cout <<"f(a)= "<<f(a)<<endl;
    cout <<"f(a+1)= "<<f(a+1)<<endl;
    cout <<"f(a+1)-f(a)= "<<f(a+1)-f(a)<<endl;
}
```

- Write a program that reads a string and output the number of times each lower case vowel {a,e,u,i,o} appears in it and print it out, your program must contain a function with one of its parameters as a string variable and returns the number of the times each lower case barrel appear in it

hint: The ASCII values of a, e, i, o, and u are **97, 101, 105, 111, and 117**, respectively.

example: string "ahmad" -the output should be: 2 vowels {a,a}.

In the main function prompt the user to enter a string and use the function created to print the number of times each lowercase appears in it.

## SOLUTION

```
int VowelCounter(string str)
{
    int sum = 0;

    for (int i = 0; i < str.length(); i++)
        if (str[i] == 97 || str[i] == 101 || str[i] == 105 || str[i] == 111 ||
            str[i] == 117)
            sum++;

    return sum;
}

int main()
{
    string str;
    cout << "please enter a string\n";
    cin >> str;
    cout << VowelCounter(str);
}
```

- Write the definition of void function **MultiplicationTable** that takes two integer one of the parameters represent number of rows of the multiplication table and the second parameter represents the number of the multiples in each row.

-The output after calling the function with input (3,10) should be:

```
1 2 3 4 5 6 7 8 9 10
2 4 6 8 10 12 14 16 18 20
3 6 9 12 15 18 21 24 27 30
```

-The output after calling the function with input (2,8) should be:

```
1 2 3 4 5 6 7 8
2 4 6 8 10 12 14 16
```

- in the main function prompt the user to enter two integers use the function created and print the multiplication table using the two integers as input.

## SOLUTION

```
void MultiplicationTable(int n1,int n2)
{
    for (int i=1; i<=n1; i++)
    {

        for(int j=1; j<= n2; j++)
            cout <<setw(5)<<i*j;

        cout <<endl;
    }
}

int main()
{
    int num1,num2;
    cout<<"please enter two integer numbers.\n";
    cin>>num1>>num2;

    MultiplicationTable(num1,num2);
}
```

- write a C++ function that takes one integer number (n) as a parameter. Your function should calculate and print the following series of (n)based on this equation

$$a_n = 2 \cdot n + 1$$

series:

1, 3, 5, 7, 9, 11, 13, 15, 17, ...

example: if user enters 4 the series is 1,3,5,7

- in the main function call the function created with parameter value 16.

## SOLUTION

```
void Series(int n)
{
    for (int i=1; i<=n; i++)
    {

        cout <<setw(3)<<2*i+1<<" ";

    }
}

int main()
{
    series(16);
}
```

```
}
```

- write a C++ function that takes one integer number (n) as a parameter. Your function should calculate and print the following series of (n) based on this equation

$$a_n = 3 \cdot n - 1$$

series:

2, 5, 8, 11, 14, ...

example: if user enters 3 the series is 2,5,8

- in the main function call the function created with parameter value 53.

## SOLUTION

```
void Series(int n)
{
    for (int i=1; i<=n; i++)
    {
        cout <<setw(3)<<3*i-1<<" ";
    }
}

int main()
{
    Series(53);
}
```

- Write a function name **Sum\_From\_To** that takes two integer numbers as a parameters and returns the value of the sum between the two integers .

for example: (4,7) the sum is  $4 + 5 + 6 + 7 = 22$

- In the main function call the function created with parameter value (9,4).



## SOLUTION

```
int Sum_From_To(int From,int To)
{
    int sum=0;
    for (int i= From; i<= To; i++)
        sum+=i;
    return sum;
}
int main()
{
    cout <<Sum_From_To(4,9);
}
```

- Write the implementation of a functions **SphereVolume** that takes one parameter of type integer, the perimeter represents the radius. Your function should calculate and return the volume of the sphere but also should ensure that the passed parameters are positive numbers otherwise set the negative values to 0

Hint: Volume =  $(3/4) * PI * radius$

PI should be constant variable = 3.14

- In the main function use the function **SphereVolume** to calculate the difference between the volume of two spheres with the radius =8 and the other=9 and return the difference.

## SOLUTION

```
const double PI=3.14;

double SphereVolume(int raduis)
{
    if (raduis<0)
        return 0;

    return (3.0/4.0)*PI*raduis;
}

int main()
{

    cout <<SphereVolume(9)-SphereVolume(8);
}
```

# ADDITIONAL QUESTIONS

---

- **Dr.Manar's Demo of practice exam of 2021.**

write a C++ program to:

1. Create 2 arrays of type int to store 10 integer numbers.
2. Initialize them with randomly generated values.
3. Print the content of them.
4. Find and print the minimum, maximum, and average values stored in each array.
5. Check if those arrays are equal or not then print a proper message.
6. Create a new array of size 20 and copy the content of the first two arrays on it (Merge them)
7. Print the content of the newly created array.

## SOLUTION

```
#include <iostream>
#include<cstdlib>
#include<ctime>
#include<iomanip>

using namespace std;

int main()
{

    int array1[10],array2[10];

    srand(time(0));

    for (int i=0; i<10; i++)
    {
        array1[i]=rand()%10+1;
        array2[i]=rand()%10+1;
    }

    cout <<"elements of the first array\n";
    for (int i=0; i<10; i++)
    {
        cout <<setw(3)<< array1[i];
    }

    cout <<"\n*****" <<endl;

    cout <<"elements of the second array\n";
    for (int i=0; i<10; i++)
    {
```

```

        cout <<setw(3)<< array2[i];

    }

    cout <<"\n*****" <<endl;

    int minimum=array1[0],maximum=array1[0];

    for (int i=0; i<10; i++)
        if(maximum<array1[i])
            maximum =array1[i];

    cout<<"the max number in array 1 is:"<<maximum<<endl;

    for (int i=0; i<10; i++)
        if(minimum>array1[i])
            minimum =array1[i];

    cout<<"the min number in array 1 is:"<<minimum<<endl;

    cout <<"\n*****" <<endl;

    minimum=array2[0],maximum=array2[0];

    for (int i=0; i<10; i++)
        if(maximum<array2[i])
            maximum =array2[i];

    cout<<"the max number in array 2 is:"<<maximum<<endl;

    for (int i=0; i<10; i++)
        if(minimum>array2[i])
            minimum =array2[i];

    cout<<"the min number in array 2 is:"<<minimum<<endl;

    cout <<"\n*****" <<endl;

    int sum=0;
    double avg;
    for (int i=0; i<10; i++)
        sum+=array1[i];

    avg=sum/10.0;

    cout <<"the sum of elements of array1 is:"<<sum<<endl;
    cout <<"the avg of elements of array1 is:"<<avg<<endl;

    cout <<"\n*****" <<endl;

    sum=0;
    avg=0;

    for (int i=0; i<10; i++)
        sum+=array2[i];

```

```

avg=sum/10.0;

cout <<"the sum of elements of array2 is:"<<sum<<endl;
cout <<"the avg of elements of array2 is:"<<avg<<endl;

cout <<"\n*****" <<endl;

bool equal=true;
for (int i=0; i<10; i++)
    if(array1[i] != array2[i] )
    {
        equal =false;
        break;
    }

if (equal)
    cout <<"The arrays are equal\n";
else
    cout <<"The arrays are not equal\n";

cout <<"\n*****" <<endl;

int array_merged[20];

for(int i=0; i<10; i++)
    array_merged[i]=array1[i];

int counter=0;
for(int i=10; i<20; i++)
{
    array_merged[i]=array2[counter];
    counter++;
}

cout <<"elements of the merged array\n";

for(int i=0; i<20; i++)
    cout <<setw(3)<<array_merged[i];

cout<<endl;
}

```

### • Dr.Amani's assignment of 2024

Write a C++ program which do the following:

1. defines a CONSTANT variable **ASIZE** = 20.
2. defines a double array **GRADES** of size **ASIZE**.
3. write a function which inputs the data into the array with the following title **fillArray**(double array[], int size)

4. write a function which prints the array with the following title **printArray** (double array[] , int size).
5. write a function which calculates the array average the following title **ArrayAverage**(double array[] , int size).
6. write a function which finds the minimum and maximum elements of the array **ArrayMinMax**(double array[] , int size, double& minimum, double& maximum).
7. In the main function, after defining the array, call the above functions to do the following: fill the array with data, print the array, find the array average, and finds the minimum and maximum elements of the array.

## SOLUTION

```
#include <iostream>
#include<cstdlib>
#include<ctime>
#include<iomanip>

using namespace std;

const int ASIZE = 20;

void fillArray(double array[], int size)
{
    for (int i = 0; i < size; i++)
    {
        cout << "enter element [" << i + 1 << "]\n";
        cin >> array[i];
    }
}

void printArray(double array[], int size)
{
    for (int i = 0; i < size; i++)

        cout << setw(3) << array[i];

    cout << endl;
}

double ArrayAverage(double array[], int size)
{
    double sum = 0;

    for (int i = 0; i < size; i++)
        sum += i;

    return sum / size;
}

void ArrayMinMax(double array[], int size, double& minimum, double& maximum)
```

```
{
    minimum = array[0];
    maximum = array[0];

    for (int i = 0; i < size; i++)
        if (minimum > array[i])
            minimum = array[i];

    for (int i = 0; i < size; i++)
        if (maximum < array[i])
            maximum = array[i];
}

int main()
{
    double GRADES[ASIZE];

    fillArray(GRADES, ASIZE);
    printArray(GRADES, ASIZE);

    cout << "the avg of this array elements =" << ArrayAverage(GRADES,ASIZE) <<
endl;

    double minimum, maximum;

    ArrayMinMax(GRADES, ASIZE, minimum, maximum);

    cout << "the minimum value in the array is:" << minimum << endl;
    cout << "the maximum value in the array is:" << maximum << endl;
}
```