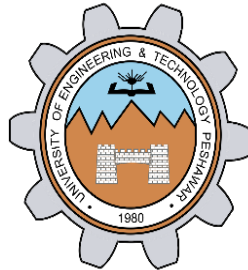


LAB # 6



Spring 2020

CSE102L Computer Programming Lab

Submitted by: **Muhammad Ali**

Registration No. : **19PWCSE1801**

Class Section: A

Submitted to:

Engr. Abdullah Hamid

(June 26, 2020)

Department of Computer Systems Engineering
University of Engineering and Technology, Peshawar

1. Write a program that takes marks as input and displays the grade using function.

```
#include<iostream>

using namespace std;

int Getgrade( int );

main(){

    int marks;

    cout<<"Enter total number out of 100"<<endl;

    cin>>marks;

    Getgrade(marks);

    return 0;

}

int Getgrade(int num )

{

char grade;

    if(num>=90 && num<=100)

    {

        grade = 'A';

    }

    else if(num>=70 && num<90){
```

```
        grade = 'B';

    }

    else if(num>=60&& num<70){

        grade = 'C';

    }


    else if(num>=50 && num<60){

        grade = 'D';

    }

    else if (num>=40 && num<50){

        grade = 'E';

    }

    else if(num<40){

        grade='f';

    }

    cout<<"Grade = "<<grade;

}
```

- 2. Write a function minmax() that takes four integers as input and display the minimum and maximum number.**

```
#include<iostream>

using namespace std;

void findmaximani(int,int,int,int);

main(){

    int n1,n2,n3,n4;

    cout<<"Enter four number ";

    cin>>n1>>n2>>n3>>n4;

    findmaximani(n1,n2,n3,n4);

    return 0;

}

void findmaximani(int a,int b,int c,int d){

    int maxi=a,mani=a;

    int maximani[4]={a,b,c,d};
```

```

for (int i=0; i<4; i++){

    if (maximani[i]>maxi)

maxi =maximani[i];


    if(maximani[i]<mani)

        mani =maximani[i];

}

cout<<"Maximum number is :"<<maxi;

cout<<"\nMinimum number is :"<<mani;

}

```

- 3. Your program should have a function named ‘prime’ which accepts an integer and return a Boolean (a true if the number is prime and false otherwise).**

```

#include<iostream>
#include<conio.h>
#include<cmath> //A header file
using namespace std;
bool prime(int num);
int main() {

    int num;

```

```

    cout<<"Enter a number: ";
    cin>>num;
    cout<<num<<" is a "<<((prime(num))?"prime":"composite")<<" number.";
    getch();
    return 0;
}

bool prime(int num)
{
    int sqroot = (int) sqrt(num); //sqrt() is defined in cmath
    while(sqroot>1) {

        float div = (float) num / sqroot; //divide number entered by user by current value
of sqroot
        if(div == floor(div)) { //and if result doesn't contain any fractional part, then
number was completely divided
            return false;
        }
        sqroot--;
    }
    return true;
}

```

- 4. Write a program to find a factorial of user input number. Use function to find factorial.**

```

#include<iostream>

using namespace std;

```

```
long factorial(int);
```

```
main(){
```

```
    int number;
```

```
    cout<<"Enter a positive  number "<<endl;
```

```
    cin>>number;
```

```
    cout<<number<<"! ="<<factorial(number);
```

```
    return 0;
```

```
}
```

```
    long factorial(int n)
```

```
    {
```

```
        int c;
```

```
        int result=1;
```

```
        for( int c=1; c<=n; c++)
```

```
            result =result*c;
```

```
        return result;
```

```
    }
```

5. **Given an integer number; you have to find the total number of minimum bit(s) which can be used to store given integer number. Implement the program using function. Function will take the integer as input and return the number of bits required.**

```
#include<iostream>
#include<conio.h>
using namespace std;
int minbit(int num);
int main() {

    int n;
    cout<<"Enter a number: ";
        cin>>n;
    cout<<"The minimum bits required for "<<n<<" are "<<minbit(n);
    getch();
    return 0;
}
int minbit(int num) {
    int min = 0;
```



```

do {
    num>>=1;
    min++;
}while(num>0);
return min;
}

```

- 6. Write a program to find the roots of a quadratic equation of type $a.x^2+b.x+c$ where a is not equal to zero.**

```

#include<iostream>
#include<conio.h>
#include<cmath> //Because we will need to use sqrt()
using namespace std;
void quadRoots(int a, int b, int c);
int main() {
    int a, b, c;
    cout<<"Enter value for coefficient of x square: ";
    cin>>a;
    if(a == 0) {
        cout<<"Invalid Input: Coefficient of x square cannot be 0";
        getch();
        return 0;
    }
    cout<<"Enter value for coefficient of x: ";
    cin>>b;
    cout<<"Enter value of constant: ";
    cin>>c;
}

```

```

    quadRoots(a, b, c);
    getch();
    return 0;
}

void quadRoots(int a, int b, int c) {

    int disc = b * b - 4 * a * c
    if(disc < 0) {
        cout<<"Roots are not real and hence cannot be computed.";
        return; //This will terminate the function
    }

    if(disc == 0) {

        float x = (float) (-1 * b) / (2 * a);
        cout<<"Roots are real and identical and are equal to "<<x;
    }else {

        float sqrtsdisc = sqrt(disc), x1 = (float) ((-1 * b) - sqrtsdisc) / (2 * a), x2 = ((-1 * b) + sqrtsdisc) / (2 *
a);

        cout<<"Real and unequal roots are "<<x1<<" and "<<x2;

    }
}

```

- 7. Write Program to compute Sin(x) using Taylor series approximation given by**

$$\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots$$
Compare the result with the built-in Library function and print both the results.

```

#include<iostream>
#include<conio.h>
#include<cmath>
using namespace std;
int fact(int n);
double expon(double base, int exp);
double Sin(double rad); //S' is capitalize to avoid name clash with built-in sin()
int main() {
    int deg;
    const double PI = 3.14159265358979323846, multple = PI / 180;
    double rad;
    cout<<"Enter value of Angle in degrees: ";
    cin>>deg;
    deg %= 360; //Degrees wrap arround 360
    rad = deg * multple;
    cout<<"Value of Built-in \"sin\" function at "<<deg<<" degrees is "<<sin(rad)<<" while our
function returned "<<Sin(rad);
    getch();
    return 0;
}
double Sin(double rad) {
    double res = rad;
    int op = -1;
    for(int i = 3; i < 15; i+=2, op*=-1) {
        double term = expon(rad, i) / fact(i);
        term *= op;
        res += term;
    }
    return res;
}

```

```
}
```

```
int fact(int n) {  
    if(n == 0) return 1;  
    return n * fact(n - 1);  
}
```

```
double expon(double base, int exp) {  
    double raised = 1;  
    while(exp >= 1) {  
        raised *= base;  
        exp--;  
    }  
    return raised;  
}
```