Submitted By:- Ravi Pandey(075BCT065)

1. Write a program to implement the following two functions on string using pointer:

   i. To calculate length of given string
   ii. To concatenate two strings.

```
#include <iostream>

using namespace std;

int strLeng(const char * str)
{
    int l=0;
    while(str[l]!='\0')
    {
        l++;
    }
    return l;
}
void concatStr(const char * str1,const char* str2)
{
    int l1 = strLeng(str1);
    int l2=strLeng(str2);
    char *res;
    res = new char[l1+l2];
    for(int i=0;i<=l1;i++)
    res[i]=str1[i];
    for(int i=0;i<=l2;i++)
    res[l1+i]=str2[i];
    cout<<res;
}
int main()
{
    concatStr("DATA","STRUCTURE");
    return 0;
}
```

OUTPUT:-

```
DATASTRUCTURE
Process returned 0 (0x0)   execution time : 2.828 s
Press any key to continue.
```

2. WAP to create a structure DATE with its member: day, month and year. Write a function which takes two pointer variables of type DATE to calculate the age of person (i.e. pass structure by reference).

Program Code

```
#include <iostream>

using namespace std;

struct DATE
{
  private:
  int year;
  int month;
  int day;
  public:
  DATE(int a=2020,int b=8,int c=31)
  {
      year=a;
      month=b;
      day=c;
  }

friend int calcAge(DATE *,DATE *);

};

int calcAge(DATE * birthYear,DATE* currYear)
{
    int yr,mn,d;
        yr=currYear->year-birthYear->year;
        if(yr<0)
        return -1;
        if(birthYear->month>=currYear->month)
            yr--;
    return yr;


}

void showAge(int age)
{
    if(age==-1)
    {
        cout<<"\nInvalid dates entered.";
    }
    else
```

```
    {
        cout<<"\nYou are "<<age<<" years old.";
    }
}
int main()
{
    DATE D1(2000,2,16);
    DATE D2(2020,3,14);
    DATE D3(2005,7,23);
    int age;
    age=calcAge(&D1,&D2);
    showAge(age);
    showAge(calcAge(&D2,&D3));
    showAge(calcAge(&D3,&D2));

}
```

Output:-

```
You are 20 years old.
Invalid dates entered.
You are 14 years old.
Process returned 0 (0x0)   execution time : 2.600 s
Press any key to continue.
```

3. WAP to swap the values (of type int, double, char) using function template.

```
#include <iostream>

using namespace std;

template <class T>
void swAp(T *a, T *b)
{
    T temp;
    temp=*a;
    *a=*b;
    *b=temp;
}
```

```
int main()
{
    int a=1,b=2;
    double c=5.1,d=6.34;
    char e='e',f ='f';
    cout<<"The initial data before swapping are:\nInteger\na = "<<a<<" b="<<b;
    cout<<"\nDouble\nc = "<<c<<" d="<<d;
    cout<<"\nChar\ne = "<<e<<" f = "<<f<<endl;
    swAp(&a,&b);
    swAp(&c,&d);
    swAp(&e,&f);
    cout<<"After swapping:\nIntegers\na = "<<a<<" b = "<<b;
    cout<<"\nDouble\nc = "<<c<<" d = "<<d;
    cout<<"\nChar\ne = "<<e<<"f = "<<f;
}
```

Output:-

```
The initial data before swapping are:
Integer
a = 1 b=2
Double
c = 5.1 d=6.34
Char
e = e f = f
After swapping:
Integers
a = 2 b = 1
Double
c = 6.34 d = 5.1
Char
e = ff = e
Process returned 0 (0x0)   execution time : 2.773 s
Press any key to continue.
```

4. Perform no.3 using class template.

```
#include <iostream>

using namespace std;

template<class T>
class Swap
{
    T a;
    T b;
    public:
```

```
    Swap(T m1=0,T m2=0)
    {
        a=m1;
        b=m2;
    }
    void showProp()
    {
        cout<<"\nA = "<<this->a<<"\tB = "<<b;
    }
    void swapProp()
    {
        T temp;
        temp =a;
        a=b;
        b=temp;
    }
};

int main()
{
    Swap<int> i(5,3);
    Swap<double> d(3.3222,545.3212);
    Swap<char> c('a','b');
    cout<<"\nBefore swapping:";
    i.showProp();
    d.showProp();
    c.showProp();

    i.swapProp();
    d.swapProp();
    c.swapProp();

    cout<<"\nAfter swapping:";
    i.showProp();
    d.showProp();
    c.showProp();
}
```

Output:-

```
Before swapping:
A = 5   B = 3
A = 3.3222      B = 545.321
A = a   B = b
After swapping:
A = 3   B = 5
A = 545.321     B = 3.3222
A = b   B = a
Process returned 0 (0x0)   execution time : 2.588 s
Press any key to continue.
```

5. Write a C++ program to perform arithmetic operations on two numbers and throw an exception if the dividend is zero or does not contain an operator. Enter the input as a+b, where ‘a’ and ‘b’ are input numbers and ‘+’ as operator. Check for the valid operators and perform the different operations like addition, subtraction, multiplication and division accordingly.

- *Without Exception Class*

```
#include <iostream>

using std::cin;
using std::cout;
template <class T>
class ArithmeticOps
{
    public:
    static T operate(T term1,T term2,char op)
    {
        switch(op)
        {
            case '+':
                return term1+term2;
            case '-':
                return term1-term2;
            case '*':
                return term1*term2;
            case '/':
                return term1/term2;
        }
    }
};
int main()
{
    float val1,val2,res;
    char op;
    cout<<"\nEnter Operation:";
    cin>>val1;
    cin>>op;
    cin>>val2;
    if(op=='/' && val2==0)
        cout<<"\nCannot divide by zero!";
    else if(op=='+' || op=='-' || op=='*' || op=='/')
    {
        ArithmeticOps<float> Ar;
        res=Ar.operate(val1,val2,op);
        cout<<"\nThe result is: "<<res;
    }
```

```
    else
        cout<<"\nINVALID OPERATOR!";
    return 0;
}
```

Output:-

```
Enter Operation:5+4

The result is: 9
Process returned 0 (0x0)   execution time : 9.762 s
Press any key to continue.
```

- *With Exception Class*

```
#include <iostream>

using namespace std;


class ArithmeticOps
{
    public:
    class ZERO{};
    class OPERR{};
    static float operate(float term1,float term2,char op)
    {
        switch(op)
        {
            case '+':
                return term1+term2;
            case '-':
                return term1-term2;
            case '*':
                return term1*term2;
            case '/':
            {
                if(term2==0)
                throw(ZERO());
                return term1/term2;
            }
            default:
                throw(OPERR());

        }
    }
};
```

```
int main()
{

    float val1,val2,res;
    char op;
    cout<<"Enter Operation: ";
    cin>>val1;
    cin>>op;
    cin>>val2;
    try
    {
        res=ArithmeticOps::operate(val1,val2,op);
        cout<<"\nThe result is: "<<res;
    }
    catch(ArithmeticOps::ZERO)
    {
        cout<<"Can't divide by Zero!";
    }
    catch(ArithmeticOps::OPERR)
    {
        cout<<"Invalid Operator!";
    }
}
```

Output:-

```
Enter Operation: 5/0
Can't divide by Zero!
Process returned 0 (0x0)   execution time : 9.811 s
Press any key to continue.
```