# DATA STRUCTURES LAB

**LAB RECORD**

***Submitted by***

**Tanishq Agarwal**

**Submitted to: Dr. Kunj Bihari Meena**



2022-2023

**Department of Computer Science & Engineering JAYPEE UNIVERSITY OF ENGINEERING & TECHNOLOGY,**

**AB ROAD, RAGHOGARH, DT. GUNA-473226 MP, INDIA**

**Table of Contents**

|  |  |
| --- | --- |
| **Lab Exercise with topic** | **Page No.** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

# Lab Exercise 1: Revisit to C++

## Q1. WAP to find out largest element of an array.

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//This program is developed by Tanishq Agarwal(211B326)

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Solution:

#include<iostream>

using namespace std;

int main(){

cout<<"\*\*TO FIND LARGEST ELEMENT OF AN ARRAY\*\*"<<endl;

cout<<"Enter original array to be sorted:"<<endl;

int \*arr = NULL;

int n=0;

cout<<"Enter number of elements in the array:"<<endl;

cin>>n;

arr= new int[n];

for(int i=0;i<n;i++)

{cin>>arr[i];}

cout<<"Array inserted successfully!!"<<endl;

cout<<"Starting operation!!"<<endl;

int large= arr[0];

for(int j=0;j<n;j++){

if(arr[j]<arr[j+1]){

large=arr[j+1];

}

else{}

}

cout<<"The largest number in the array is:"<<endl;

cout<<large;

return 0;

}

## Q2. WAP to search an element in array.

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//This program is developed by Tanishq Agarwal(211B326)

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Solution:

#include<iostream>

using namespace std;

int main(){

cout<<"Enter original array data from which data set is to be searched!!"<<endl;

int \*arr= NULL;

int n;

cout<<"Enter number of elements in the original array!"<<endl;

cin>>n;

arr= new int[n];

cout<<"Enter array elements:"<<endl;

for (int i = 0; i < n; i++)

{

cin>>arr[i];

}

int query;

cout<<"Enter the value of data to be searched for:"<<endl;

cin>>query;

cout<<"SEARCHING STARTED!!"<<endl;

for (int i = 0; i < n; i++)

{

if(arr[i]==query){

cout<<"Data found and query matched in the array at position:"<<i<<endl;

goto ex;

}

else{

}

}

cout<<"The data query searched is not found in the array!!"<<endl;

ex:

break;

return 0;

}

## Q3. WAP to check whether the number is prime or not.

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//This program is developed by Tanishq Agarwal(211B326)

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Solution:

#include<iostream>

using namespace std;

int main(){

int num=0;

int fact=0;

cout<<"Enter the number for being checked if it is prime or not>>:"<<endl;

cin>>num;

for (int i = 2; i <= sqrt(num); i++)

{

if(num%i==0)

{fact++;}

else{}

}

if(fact==1){

cout<<"The number is prime!!"<<endl;

}

else{

cout<<"The number is not prime!!"<<endl;

}

return 0;

}

## Q4. WAP to calculate xy where x and y are two integer numbers entered by the user. [do not

## use pow() function].

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//This program is developed by Tanishq Agarwal(211B326)

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Solution:

#include<iostream>

using namespace std;

int main(){

int x,y=0;

int pow=1;

cout<<"Enter base x:"<<endl;

cin>>x;

cout<<"Enter power for base y:"<<endl;

cin>>y;

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

pow=x\*x;

}

cout<<"Value of x^y is :"<<pow<<endl;

return 0;

}

## Q5. WAP to replace a character by another character in a string. Take both the choices from

## the user.

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//This program is developed by Tanishq Agarwal(211B326)

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Solution:

#include<iostream>

using namespace std;

int main(){

char \*str\_orig;

int n=0;

char ch\_rep;

char ch\_replacer;

cout<<"Enter the string length"<<endl;

cin>>n;

str\_orig=new char[n+1];

cout<<"Enter a string first to start the question:"<<endl;

cin>>str\_orig;

cout<<"Enter the character to be replaced form the string:"<<endl;

cin>>ch\_rep;

cout<<"Enter the character to be replaced with:"<<endl;

cin>>ch\_replacer;

cout<<"Starting the replacement"<<endl;

for (int i = 0; i < n; i++)

{

if(str\_orig[i]==ch\_rep){

str\_orig[i]=ch\_replacer;

}

else{}

}

cout<<"The edited string with replaced character is :"<<str\_orig<<endl;

return 0;

}

## Q6. WAP to find the reverse of given string.

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//This program is developed by Tanishq Agarwal(211B326)

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Solution:

#include<iostream>

using namespace std;

int main(){

int sz=0;

int n=0;

char \*str\_orig;

cout<<"Enter the string length to be enetered:"<<endl;

cin>>n;

str\_orig=new char[n+1];

cout<<"Enter the string to be reversed:"<<endl;

cin>>str\_orig;

cout<<"Starting reversal process:"<<endl;

char \*str\_rev;

str\_rev= new char[n+1];

sz=n+1;

for (int i = 0; i<=sz; i++)

{

str\_rev[i]=str\_orig[n+1];

n--;

}

cout<<"The reversed string is :"<<str\_rev<<endl;

return 0;

}

## Q7. WAP to sort the array and ask the choice from user for ascending/descending.

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//This program is developed by Tanishq Agarwal(211B326)

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Solution:

#include<iostream>

using namespace std;

int main(){

int n=0;

cout<<"Enter no of elements in the array to be sorted:"<<endl;

cin>>n;

int \*arr;

arr= new int[n];

cout<<"Enter elements in the original array unsorted:"<<endl;

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

cin>>arr[i];

}

cout<<"Array elements entered successfully!!"<<endl;

char choice;

C:

cout<<"Enter the type of sorting to be performed:"<<endl<<"1.) 'A' or 'a' for ascending order."<<endl<<"2.) 'D' or 'd' for descending order."<<endl;

cin>>choice;

if(choice=='A' || choice=='a')

{

cout<<"You choose Ascending sorting of the array;"<<endl;

int min=arr[0];

for (int i = 0; i < n; i++)

{

if(arr[i]<min){

min=arr[i];

}

}

}

else if(choice=='D' || choice=='d'){

}

else{

cout<<"Enter a valid choice again!!"<<endl;

goto C;

}

return 0;

}

## Q8. WAP to find a word in given statement.

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//This program is developed by Tanishq Agarwal(211B326)

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Solution:

#include<iostream>

using namespace std;

int main(){

return 0;

}

## Q9. WAP to concatenate two strings using pointer.

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//This program is developed by Tanishq Agarwal(211B326)

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Solution:

#include<iostream>

using namespace std;

int main(){

return 0;

}

## Q10. WAP to create a dynamic array of user desired size and search an element in that array.

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//This program is developed by Tanishq Agarwal(211B326)

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Solution:

#include<iostream>

using namespace std;

int main(){

int n;

cout<<"Enter the size of array you want to create:"<<endl;

cin>>n;

int \*array;

array=new int[n];

cout<<"Array of size "<<n<<" generated successfully!!"<<endl;

int srch;

cout<<"Enter the element to be searched for:"<<endl;

cin>>srch;

for (int i = 0; i <n; i++)

{

if(array[i]==srch){

cout<<"The element is found successfully at position:"<< n<<endl;

exit(0);

}

else{

}

}

cout<<"The element requested for search operation was not found in the array elemnts provided!!"<<endl;

return 0;

}

**Advanced Programming Problems:**

## Q11. WAP to calculate difference between two time periods using the C structures.

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//This program is developed by Tanishq Agarwal(211B326)

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Solution:

#include<iostream>

using namespace std;

struct time

{

int hr=0;

int min=0;

int sec=0;

};

int main(){

time t1;

time t2;

cout<<"Enter time 1 hours:"<<endl;

cin>>t1.hr;

cout<<"Enter time 1 minutes:"<<endl;

cin>>t1.min;

cout<<"Enter time 1 seconds:"<<endl;

cin>>t1.sec;

cout<<"Enter time 2 hours:"<<endl;

cin>>t2.hr;

cout<<"Enter time 2 minutes:"<<endl;

cin>>t2.min;

cout<<"Enter time 2 seconds:"<<endl;

cin>>t2.sec;

cout<<"Time data entered is:"<<t1.hr<<":"<<t1.min<<":"<<t1.sec<<"and "<<t2.hr<<":"<<t2.min<<":"<<t2.sec<<"respectively."<<endl;

int hr\_diff,min\_diff,sec\_diff=0;

if(t1.hr>t2.hr){

hr\_diff=t1.hr-t2.hr;

}

else{

hr\_diff=t2.hr-t1.hr;

}

if(t1.min>t2.min){

min\_diff=t1.min-t2.min;

}

else{

min\_diff=t2.min-t1.min;

}

if(t1.sec>t2.sec){

sec\_diff=t1.sec-t2.sec;

}

else{

sec\_diff=t2.sec-t1.sec;

}

cout<<"The time difference between entered time is: "<<hr\_diff<<":"<<min\_diff<<":"<<sec\_diff<<endl;

return 0;

}

## Q12. WAP to add two complex numbers by passing structure to a function.

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//This program is developed by Tanishq Agarwal(211B326)

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Solution:

#include<iostream>

using namespace std;

struct complex

{

int real=0;

int imag=0;

};

complex compadd(complex comp1,complex comp2){

complex res;

res.real=comp1.real+comp2.real;

res.imag=comp1.imag+comp2.imag;

return res;

}

int main(){

complex c1;

complex c2;

complex c\_res;

cout<<"Enter complex number 1 real part and imaginary part respectively:"<<endl;

cin>>c1.real>>c1.imag;

cout<<"Enter complex number 2 real part and imaginary part respectively:"<<endl;

cin>>c2.real>>c2.imag;

c\_res=compadd(c1,c2);

cout<<"The additive result of the two complex numbers provided is:"<<c\_res.real<<"+"<<c\_res.imag<<endl;

return 0;

}

# Lab Exercise 2: Revisit to C++

## Q1. WAP to generate a Fibonacci series up to n terms.

## Input

## Input number of terms: 10

## Output

## Fibonacci series:

## 0, 1, 1, 2, 3, 5, 8, 13, 21, 34

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//This program is developed by Tanishq Agarwal(211B326)

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Solution:

#include<iostream>

using namespace std;

int main(){

int n=1;

cout<<"Enter the number of fibonacci series you want to print:"<<endl;

cin>>n;

int s=0,f=1;

cout<<"Fibonacci series is :"<<s<<","<<f<<",";

int nxt\_sm=0;

for(int i=2;i<=n-2;i++){

nxt\_sm=f+s;

s=f;

f=nxt\_sm;

cout<<nxt\_sm<<",";

}

return 0;

}

## Q2. WAP to find out series sum of 1^2 + 2^2 + …. + n^2

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//This program is developed by Tanishq Agarwal(211B326)

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Solution:

#include<iostream>

using namespace std;

int main(){

int n=1;

int sum=0;

cout<<"Enter the series limit upto n:"<<endl;

cin>>n;

cout<<"The value of the series is:";

for (int i = 1; i < n; i++)

{

sum+=i\*i;

}

cout<<"The sum upto n for given series is:"<<sum<<endl;

return 0;

}

## Q3. WAP to find out GCD of two numbers.

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//This program is developed by Tanishq Agarwal(211B326)

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Solution:

#include<iostream>

#include<math.h>

using namespace std;

int main(){

int a=0,b=0;

cout<<”Enter a and b :”<<endl;

cin>>a>>b;

int result = min(a, b);

while (result > 0) {

if (a % result == 0 && b % result == 0) {

break;

}

result--;

}

cout<<”The GCD of entered a and b is :”<<result;

return 0;

}

## Q4. WAP to multiply two numbers by using addition.

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//This program is developed by Tanishq Agarwal(211B326)

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Solution:

#include<iostream>

using namespace std;

int main(){

int a,b=0;

int sum=0;

cout<<"Enter the numbers to be multiplied using addition:"<<endl;

cin>>a,b;

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

sum+=a;

}

cout<<"The value of a\*b using additive multiplication is:"<<sum<<endl;

return 0;

}

## Q5. WAP to convert a binary number into decimal.

## /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//This program is developed by Tanishq Agarwal(211B326)

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Solution:

#include<iostream>

using namespace std;

int main(){

int sz=0;

cout<<"Enter the size of the binary number in digits:"<<endl;

cin>>sz;

int \*bin;

bin = new int[sz];

cout<<"Enter binary number:"<<endl;

for(int j=0;j<sz;j++){

cout<<"Enter the binary value for position:"<<j<<endl;

cin>>bin[j];

}

int dec=0;

for (int i = sz; i >= 0; i--)

{

dec+=bin[i]\*pow(2,sz-i);

}

cout<<"The decimal value for entered binary number is:"<<dec<<endl;

return 0;

}

## Q6. WAP to convert a decimal into binary number.

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//This program is developed by Tanishq Agarwal(211B326)

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Solution:

#include<iostream>

using namespace std;

int main(){

int dec=0;

int bin[20]={0};

cout<<"Enter the decimal number to be converted into binary format:"<<endl;

cin>>dec;

int i=0;

int temp\_bin[20];

while(dec!=0){

temp\_bin[i]=dec%2;

dec=dec/2;

i++;

}

for (int i = 0; i < 20; i++)

{

bin[i]=temp\_bin[19-i];

}

cout<<"The binary equivalent of given decimal number is:"<<endl;

for (int i = 0; i < 20; i++)

{

cout<<bin[i];

}

return 0;

}

## Q7. WAP to display lower triangular matrix of a given n by n size matrix entered by user.

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//This program is developed by Tanishq Agarwal(211B326)

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Solution:

#include<iostream>

using namespace std;

int main(){

int i, j, rows, columns, a[10][10];

cout<<” Enter Number of rows and columns of the matrix to be entered : ";

cin>>i>>j;

cout<<”Enter matrix elements ";

for(rows = 0; rows < i; rows++)

{

for(columns = 0;columns < j;columns++)

{

cin>>a[rows][columns];

}

}

for(rows = 0; rows < i; rows++)

{

cout<<endl;

for(columns = 0; columns < j; columns++)

{

if(rows >= columns)

{

cout<<a[rows][columns];

}

else

{

printf("0 ");

}

}

}

return 0;

}

## Q8. WAP to find out nCr factor of given numbers.

## Note:

## nCr = n! /((n-r)!r!)

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//This program is developed by Tanishq Agarwal(211B326)

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Solution:

#include<iostream>

using namespace std;

int main(){

int n,r=0;

int n\_fact=1;

int r\_fact=1;

int nr\_fact=1;

cout<<"Enter the value of n and r:"<<endl;

cin>>n,r;

//Calculating n!

for (int i = 1; i <= n; i++)

{

n\_fact\*=i;

}

//Calculating r!

for (int i = 1; i <= r; i++)

{

r\_fact\*=i;

}

//Calculating (n-r)!

for (int i = 1; i <= (n-r); i++)

{

nr\_fact\*=i;

}

//Calculating value for nCr

int res=n\_fact/(nr\_fact\*r\_fact);

cout<<"The value of combinational factor nCr for given n and r is:"<<res<<endl;

return 0;

}

**Advanced Programming Problems:**

## Q9. WAP for finding the element which appears maximum number of times in

## the array.

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//This program is developed by Tanishq Agarwal(211B326)

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Solution:

#include<iostream>

using namespace std;

int main(){

int maxcnt=0;

int max\_freq\_elem=0;

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

int count=0;

for(int j=0;j<n;j++){

if(arr[i] == arr[j])

count++;

}

if(count>maxcnt){

maxcnt=count;

max\_freq\_elem = arr[i];

}

}

return 0;

}

## Q10. Consider that you are given with a database of employee records (at least 5).

## Each employee record having following information –

## Emp\_id(integer), Emp\_name(string), Emp\_city(string)

## Assume that Emp\_id is unique. Write a function for taking database and put

## it in your header file. Use this function by including your own header file for

## following questions.

## {Use the structure for creating database}a. Write a function to find out the employee record from this database on

## the basis of Emp\_id.

## b. Write a function to sort the employee records on the basis of Emp\_id.

## c. Write a function to sort (alphabetically) the array of characters.

## d. Write a function to count the number of employees in database.

## e. Write a function to add 5 more records in database.

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//This program is developed by Tanishq Agarwal(211B326)

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Solution:

**Datab.h:**

//datab.h

#pragma once

#include<iostream>

namespace datab

{

struct emp\_data{

int Emp\_id=0;

string Emp\_name;

string Emp\_city;

};

struct emp\_data ed[100];

void emp\_entry(int n){

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

cout<<"Enter employee number for Employee number:"<<i+1<<endl;

cin>>ed[i].Emp\_id;

cout<<"Enter employee name for Employee number:"<<i+1<<endl;

cin>>ed[i].Emp\_name;

cout<<"Enter employee city for Employee number:"<<i+1<<endl;

cin>>ed[i].Emp\_city;

}

cout<<"Data entered successfully!!"<<endl;

}

}

**Main.cpp:**

#include<iostream>

using namespace std;

#include”datab.h” //Header File Imported in Main.cpp file

using namespace datab;

//Functions

void search\_emp(int n,int k){

cout<<"Searching for Emp\_ID:"<<n<<endl;

for(int j=0;j<k;j++){

if(datab::ed[j].Emp\_id==n){

cout<<"Data found!!"<<endl;

cout<<"Data: Employee ID: "<<datab::ed[j].Emp\_id<<" Employee name: "<<datab::ed[j].Emp\_name<<" Employee city: "<<datab::ed[j].Emp\_city<<endl;

goto V;

}

else{}

}

cout<<"Data not found for given Employee ID!!"<<endl;

V:{}

}

void emp\_sort(int l){

cout<<"Sorting started for given Emp\_ID's in the database:"<<endl;

int max\_id=ed[0].Emp\_id;

int temp\_emp\_id=0;

string temp\_emp\_name;

string temp\_emp\_city;

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

if(ed[i].Emp\_id>=max\_id){

temp\_emp\_id=ed[i].Emp\_id;

ed[i].Emp\_id=ed[i+1].Emp\_id;

ed[i+1].Emp\_id=temp\_emp\_id;

temp\_emp\_name=ed[i].Emp\_name;

ed[i].Emp\_name=ed[i+1].Emp\_name;

ed[i+1].Emp\_name=temp\_emp\_name;

temp\_emp\_city=ed[i].Emp\_city;

ed[i].Emp\_city=ed[i+1].Emp\_city;

ed[i+1].Emp\_city=temp\_emp\_city;

}

else{}

cout<<"Data sorted successfully!!"<<endl<<"Printing sorted now:"<<endl;

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

cout<<"Data: Employee ID: "<<datab::ed[i].Emp\_id<<" Employee name: "<<datab::ed[i].Emp\_name<<" Employee city: "<<datab::ed[i].Emp\_city<<endl;

}

}

}

int count=0;

void count\_emp(){

int n=11;

while(n!=0){

if (ed->Emp\_id!=0)

{

count++;

n--;

}

else{n--;}

}

}

void add\_emp(int cnt){

cout<<"Add 5 more employees to the database:"<endl;

for (int i = cnt; i < cnt+5; i++)

{

cout<<"Enter employee number for Employee number:"<<i+1<<endl;

cin>>ed[i].Emp\_id;

cout<<"Enter employee name for Employee number:"<<i+1<<endl;

cin>>ed[i].Emp\_name;

cout<<"Enter employee city for Employee number:"<<i+1<<endl;

cin>>ed[i].Emp\_city;

}

}

int main(){

int no\_of\_emp=0;

cout<<"Enter the number of Employees in the database:"<<endl;

cin>>no\_of\_emp;

cout<<"Enter employee database values:"<<endl;

datab::emp\_entry(no\_of\_emp);

//Custom functions

int i=0;

cout<<" Enter the Employee ID of employee to be searched:";

cin>>i;

search\_emp(i,no\_of\_emp);

emp\_sort(no\_of\_emp);

count\_emp();

add\_emp();

return 0;

}

# Lab Exercise 3: Arrays

## Q1. WAP to insert new element at given index number in the array.

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//This program is developed by Tanishq Agarwal(211B326)

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Solution:

#include<iostream>

using namespace std;

int main(){

char arr\_temp[50];

char temp;

int sz=0;

for (int i = 0; i < 50; i++)

{

cin>>temp;

if(temp!=''){

arr\_temp[i]=temp;

}

else{

break;

sz=i+1;

}

}

char \*arr\_orig;

arr\_orig=new char[sz+1];

int pos=0;

cout<<"Enter the position to enter the character in the array:"<<endl;

cin>>pos;

char temp\_char;

cout<<"Enter the character to be entered in this array:"<<endl;

cin>>temp\_char;

for (int i = 0; i < sz; i++)

{

if(i<pos){

arr\_orig[i]=arr\_temp[i];

}

else if(i==pos){

arr\_orig[i]=temp\_char;

}

else{

arr\_orig[i]=arr\_temp[i-2];

}

}

cout<<"The new array with inserted value is :"<<endl;

for (int i = 0; i < sz; i++)

{

cout<<arr\_orig[i];

}

return 0;

}

## Q2. WAP to implement the linear search. Use function concept, if element is found then

## return index number of element otherwise return -1.

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//This program is developed by Tanishq Agarwal(211B326)

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Solution:

#include<iostream>

using namespace std;

int linear\_search(int \*arr,int sz, int n){

int index=0;

for (int i = 0; i < sz; i++)

{

if(arr[i]==n){return i;}

}

}

int main(){

int\* arr;

arr=new int[10];

cout<<"Enter the array elements:"<<endl;

for (int i = 0; i < count; i++)

{

cin>>arr[i];

}

cout<<"Enter the value to be searched in the array:"<<endl;

cin>>search;

int arr\_size=sizeof(arr)/sizeof(int);

int res=linear\_search(arr,arr\_size,search);

if(res<0){

cout<<"Data not Found!!"<<endl;

}

else{

cout<<"Data Found at index:"<<res;

}

return 0;

}

## Q3. WAP to delete an element from an array, use search algorithm to find the index number

## of given number; if element to be deleted is not found then print “Error: element not

## found”.

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//This program is developed by Tanishq Agarwal(211B326)

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Solution:

#include<iostream>

#include<math.h>

using namespace std;

int del(int\* arr,int n,int s){

int index=-1;

for(int i=0;i<n;i++)

if (arr[i]==s)

index=i;

for(int i=index;i<n-1;i++){

arr[i]=arr[i+1];}

arr[n-1]=0;

return index;

}

int main(){

int n=0;

cout<<"Enter the size of array:\n";

cin>>n;

int s, arr[n];

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

cin>>arr[i];}

cout<<"Enter the element to be deleted:\n";

cin>>s;

int index=del(arr,n,s);

if (index==-1)

cout<<"Error: element not found\n";

else

cout<<"Element Successfully Deleted\n";

return 0;

}

## Q4. WAP for checking whether there are any duplicated elements in the array or not?

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//This program is developed by Tanishq Agarwal(211B326)

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Solution:

#include<iostream>

using namespace std;

int dupl(int\* arr,int n){

int temp=arr[0],d=-1;

for(int i=0;i<n;i++)

for(int j=i+1;j<n;j++){

if (arr[i]==arr[j])

d=j;}

return d;

}

int main(){

int n=0;

cout<<"Enter the size of array:\n";

cin>>n;

int arr[n];

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

cin>>arr[i];}

int d=dupl(arr,n);

if (d==-1)

cout<<"No Duplicate Element Found in the array\n";

else

cout<<"There are Duplicate Element in the array at index "<<d;

return 0;

}

## Q5. Mary is a kindergarten teacher. She has given a task to the children after teaching them a list of words. The task is to find the unknown words (other than the words they already

## know) from the given text. Write a function which accepts the text and the known list of

## words and prints a set of unknown words found. If there are no unknown words found

## then prints “Successful”. [Hint use find\_word() function of Lab 1]

## /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//This program is developed by Tanishq Agarwal(211B326)

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Solution:

#include<iostream>

using namespace std;

int main(){

return 0;

}

## Q6. [This is a Q. 10 of Lab 2, Solve it if you have not done earlier] Consider that you are

## given with a data base of employee records (at least 5). Each employee record having

## following information –

## Emp\_id(integer)

## Emp\_name(string)

## Emp\_city

## Assume that Emp\_id is unique. Write a function for taking data base and put it in your

## header file. Use this function by including your own header file for following questions.

## {Use the structure for creating data base}

## a. Write a function to find out the employee record from this data base on the base

## of Emp\_id.

## b. Write a function to sort the employee records on the base of Emp\_id.

## c. Write a function to sort (alphabetically) the array of characters.

## d. Write a function to count the number of employees in data base.

## e. Write a function to add 5 more records in data base.

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//This program is developed by Tanishq Agarwal(211B326)

//The program was earlier solved as Lab 2 Q.10

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Lab Exercise 4: Arrays

# ( Hacker Rank )

## Q1. Write a C program to reverse the elements of an array.

## https://www.hackerrank.com/challenges/reverse-array-c/problem

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//This program is developed by Tanishq Agarwal(211B326)

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Solution:

#include <stdio.h>

#include <stdlib.h>

int main()

{

int num, \*arr, i;

scanf("%d", &num);

arr = (int\*) malloc(num \* sizeof(int));

for(i = 0; i < num; i++) {

scanf("%d", arr + i);

}

int temp;

for(i = 0; i < num/2; i++) {

temp=\*(arr+i);

\*(arr+i)=\*(arr+(num-1-i));

\*(arr+(num-1-i))=temp;

}

for(i = 0; i < num; i++)

printf("%d ", \*(arr + i));

return 0;

}

## Q2. Write a C program to print the frequency of the digits in given alphanumeric string.

## https://www.hackerrank.com/challenges/frequency-of-digits-1/problem

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//This program is developed by Tanishq Agarwal(211B326)

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Solution:

#include <stdio.h>

#include <string.h>

#include <math.h>

#include <stdlib.h>

int main() {

int count[2][10]={0,0};

char test\_str[50];

scanf("%s",test\_str);

for(int i=0;i<50;i++){

if(test\_str[i]>='0' && test\_str[i]<='9'){

switch(test\_str[i]){

case '0': count[1][0]++;

break;

case '1': count[1][1]++;

break;

case '2': count[1][2]++;

break;

case '3': count[1][3]++;

break;

case '4': count[1][4]++;

break;

case '5': count[1][5]++;

break;

case '6': count[1][6]++;

break;

case '7': count[1][7]++;

break;

case '8': count[1][8]++;

break;

case '9': count[1][9]++;

break;

default: break;

}

}

else{}

}

for(int i=0;i<=9;i++){

printf("%d",count[1][i]);

printf(" ");

}

return 0;

}

## Q3. Write C program to complete “Students Marks Sum’ as mentioned below:

## https://www.hackerrank.com/challenges/students-marks-sum/problem

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//This program is developed by Tanishq Agarwal(211B326)

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Solution:

#include <stdio.h>

#include <string.h>

#include <math.h>

#include <stdlib.h>

int marks\_summation(int\* marks, int number\_of\_students, char gender) {

int gs=0,bs=0;

for(int i=0;i<number\_of\_students;i+=2)

{

bs+=\*(marks+i);

}

for(int i=1;i<number\_of\_students;i+=2)

{

gs+=\*(marks+i);

}

if(gender=='g'){

return gs;}

else {

return bs;

}

}

int main() {

int number\_of\_students;

char gender;

int sum;

scanf("%d", &number\_of\_students);

int \*marks = (int \*) malloc(number\_of\_students \* sizeof (int));

for (int student = 0; student < number\_of\_students; student++) {

scanf("%d", (marks + student));

}

scanf(" %c", &gender);

sum = marks\_summation(marks, number\_of\_students, gender);

printf("%d", sum);

free(marks);

return 0;

}

## Q4. Write a C/C++ program to left rotate an array of integers by d times.

## https://www.hackerrank.com/challenges/array-left-rotation/problem

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//This program is developed by Tanishq Agarwal(211B326)

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Solution:

#include <bits/stdc++.h>

using namespace std;

string ltrim(const string &);

string rtrim(const string &);

vector<string> split(const string &);

vector<int> rotateLeft(int d, vector<int> arr) {

//int arr\_size=arr.size();

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

//rotate and update term

arr.push\_back(arr[0]);

//Update array

arr.erase(arr.begin());

}

return arr;

}

int main()

{

ofstream fout(getenv("OUTPUT\_PATH"));

string first\_multiple\_input\_temp;

getline(cin, first\_multiple\_input\_temp);

vector<string> first\_multiple\_input = split(rtrim(first\_multiple\_input\_temp));

int n = stoi(first\_multiple\_input[0]);

int d = stoi(first\_multiple\_input[1]);

string arr\_temp\_temp;

getline(cin, arr\_temp\_temp);

vector<string> arr\_temp = split(rtrim(arr\_temp\_temp));

vector<int> arr(n);

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

int arr\_item = stoi(arr\_temp[i]);

arr[i] = arr\_item;

}

vector<int> result = rotateLeft(d, arr);

for (size\_t i = 0; i < result.size(); i++) {

fout << result[i];

if (i != result.size() - 1) {

fout << " ";

}

}

fout << "\n";

fout.close();

return 0;

}

string ltrim(const string &str) {

string s(str);

s.erase(

s.begin(),

find\_if(s.begin(), s.end(), not1(ptr\_fun<int, int>(isspace)))

);

return s;

}

string rtrim(const string &str) {

string s(str);

s.erase(

find\_if(s.rbegin(), s.rend(), not1(ptr\_fun<int, int>(isspace))).base(),

s.end()

);

return s;

}

vector<string> split(const string &str) {

vector<string> tokens;

string::size\_type start = 0;

string::size\_type end = 0;

while ((end = str.find(" ", start)) != string::npos) {

tokens.push\_back(str.substr(start, end - start));

start = end + 1;

}

tokens.push\_back(str.substr(start));

return tokens;

}

# Lab Exercise 5: Searching

## Q1. Write a program to implement binary search algorithm. Assume user will enter the sorted array.

## https://www.hackerrank.com/contests/launchpad-1-winter-challenge/challenges/binary-

## search-basic

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//This program is developed by Tanishq Agarwal(211B326)

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Solution:

#include <cmath>

#include <cstdio>

#include <vector>

#include <iostream>

#include <algorithm>

using namespace std;

int binary\_search(int arr[],int key,int low,int high){

int mid=(high+low)/2;

if(key==arr[mid]){

return mid;

}

else if(key>arr[mid]){

return binary\_search(arr, key,low=mid+1,high);

}

else if(key<arr[mid]){

return binary\_search(arr,key,low,high=mid-1);

}

return -1;

}

int main() {

/\* Enter your code here. Read input from STDIN. Print output to STDOUT \*/

int n=0;

cin>>n;

int arr[n];

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

cin>>arr[i];

}

int k=0;

cin>>k;

int h=sizeof(arr)/sizeof(int);

cout<<binary\_search(arr,k,0,h-1);

return 0;

}

## Q2. Write a function which accepts an array of integers along with the size of it. The numbers are arranged in the list in increasing order until a particular index and after that it is arranged in decreasing order. This function should find and return the index position at

## which the increasing list starts decreasing. Call this function from main function.

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//This program is developed by Tanishq Agarwal(211B326)

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Solution:

#include<iostream>

using namespace std;

int arr\_inc\_dec(int arr[],int n){

for (int i = 0; i < n; i++)

{

if(arr[i]>arr[i+1]&&arr[i]>arr[i+2]){

return i+1;

}

}

return -1;

}

int main(){

int n=0;

cout<<"Enter the number of elements in the array:"<<endl;

cin>>n;

int arr[n];

cout<<"Enter an array of integers as described in question:"<<endl;

for (int i = 0; i < n; i++)

{

cin>>arr[i];

}

cout<<arr\_inc\_dec(arr,n);

return 0;

}

## Q3. Write a program to check whether given Matrix is sparse or not. We say a matrix as

## sparse when more than 50% of total elements are zero. If matrix is sparse then represent

## it in triplet form with the help of array data structure. Also print the number of bytes that

## are saved or wasted when you represent input matrix in the triplet form.

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//This program is developed by Tanishq Agarwal(211B326)

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Solution:

#include<iostream>

using namespace std;

int main()

{

int m=0;

int n=0;

cout<<"Enter size of the matrix in rows and columns (m\*n):"<<endl;

cin>>m>>n;

int inp\_matrix[m][n];

int count\_zero=0;

cout<<"Enter the values for the input matrix to be checked:"<<endl;

for (int i = 0; i < m; i++)

{

for (int j = 0; j< n; j++)

{

cin>>inp\_matrix[i][j];

if(inp\_matrix[i][j]==0){

count\_zero++;

}

}

}

if(count\_zero>((m\*n)/2)){

cout<<"Matrix entered is sparse!!"<<endl;

cout<<"The triplet representation of the given matrix will be:"<<endl;

int trip\_mat[(m\*n)-count\_zero][3];

int k=0;

for (int i = 0; i < m; i++)

{

for (int j = 0; j < n; j++)

{

if(inp\_matrix[i][j]!=0){

trip\_mat[k][0]=i;

trip\_mat[k][1]=j;

trip\_mat[k][2]=inp\_matrix[i][j];

k++;

}trip\_mat[k][3]=inp\_matrix[i][j];

}

}

for (int i = 0; i < k; i++)

{

for (int j = 0; j < 3; j++)

{

cout<<trip\_mat[i][j]<<" ";

}

cout<<endl;

}

}

else{

cout<<"Matrix entered is not sparse!!"<<endl;

return 0;

}

return 0;

}

## Q4. Write a time efficient program for finding the element which appears maximum number

## of times in the array.

## Sample input: 2, 4, 5, 6, 8, 9, 10, 13, 2, 3, 2

## Sample output: 2 [as 2 is coming three times]

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//This program is developed by Tanishq Agarwal(211B326)

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Solution:

#include<iostream>

using namespace std;

int main(){

int n=0;

cout<<"Enter the number of elements you want in an array to be enetered:"<<endl;

cin>>n;

int arr[n];

cout<<"Enter the array:"<<endl;

for (int i = 0; i < n; i++)

{

cin>>arr[i];

}

int count\_arr[2][n]={0,0};

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

count\_arr[0][i]=arr[i];

}

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

if(arr[i]==count\_arr[0][i]){

count\_arr[1][i]++;

}

}

int max\_count=count\_arr[1][0];

int element=0;

for (int i = 0; i < n; i++)

{

if(count\_arr[1][i]>max\_count){

max\_count=count\_arr[1][i];

element=count\_arr[0][i];

}

}

cout<<"Element :"<<element<<"occurred max no of times i.e.: "<<max\_count<<endl;

return 0;

}

# Lab Exercise 6: Sorting

## Q1. WAP to implement a function Rdm(n) which returns an array of random

## numbers{between 0 to 99}, where n is the size of array. (Hint: use dynamic memory

## allocation concept)

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//This program is developed by Tanishq Agarwal(211B326)

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Solution:

#include<iostream>

#include<ctime>

#include<cstdlib>

using namespace std;

int\* rdm(int n){

int\* temp\_arr=(int \*)malloc(n\*sizeof(int));

int random\_num=0;

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

random\_num=1+(rand() % 99);

//cout<<random\_num<<endl;

temp\_arr[i]=random\_num;

}

return temp\_arr;

}

int main(){

int n=0;

cout<<"Enter the number of elements to be stored in the array:"<<endl;

cin>>n;

int \*temp = rdm(n);

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

cout<<temp[i]<<endl;

}

return 0;

}

## NOTE: Use Rdm function by putting it into separate header file for questions number 2 through 6.

## Rdm.h:

## //Header file designed for Q2 to Q6:

## #include<iostream>

## namespace rdm\_name{

## int\* rdm(int n){

## int\* temp\_arr=(int \*)malloc(n\*sizeof(int));

## int random\_num=0;

## for(int i=0;i<n;i++){

## random\_num=1+(rand() % 99);

## //cout<<random\_num<<endl;

## temp\_arr[i]=random\_num;

## }

## return temp\_arr;

## }

## }

## Q2. WAP to implement the bubble sort and show the output of each pass.

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//This program is developed by Tanishq Agarwal(211B326)

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Solution:

#include<bits/stdc++.h>

#include"Rdm.h"

using namespace std;

void bubble\_sort(int\* arr,int n){

int i, j;

for (i = 0; i < n - 1; i++)

for (j = 0; j < n - i - 1; j++)

if (arr[j] > arr[j + 1])

swap(arr[j], arr[j + 1]);

}

int main(){

int n=0;

cout<<"Enter n:";

cin>>n;

int \*arr=new int(n);

arr=Rdm(n);

cout<<"Before Bubble Sort:\n";

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

cout<<arr[i]<<” ”;

}

bubble\_sort(arr,n);

cout<<"After Bubble Sort:\n";

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

cout<<arr[i]<<” ”;

}

return 0;

}

## Q3. WAP to implement the selection sort and show the output of each pas.

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//This program is developed by Tanishq Agarwal(211B326)

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Solution:

#include<bits/stdc++.h>

#include"Rdm.h"

using namespace std;

void selection\_sort(int\* arr,int n){

int i, j, min\_idx;

for (i = 0; i < n-1; i++){

min\_idx = i;

for (j = i+1; j < n; j++)

if (arr[j] < arr[min\_idx])

min\_idx = j;

swap(&arr[min\_idx], &arr[i]);

}

}

int main(){

int n=0;

cout<<"Enter n:";

cin>>n;

int \*arr=new int(n);

arr=Rdm(n);

cout<<"Before Selection Sort:\n";

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

cout<<arr[i]<<” ”;

}

selection\_sort(arr,n);

cout<<"After Selection Sort:\n";

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

cout<<arr[i]<<” ”;

}

}

## Q4. WAP to implement the insertion sort and show the output of each pass.

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//This program is developed by Tanishq Agarwal(211B326)

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Solution:

#include<bits/stdc++.h>

#include"Rdm.h"

using namespace std;

void Insertion\_sort(int\* arr,int n){

int i, key, j;

for (i = 1; i < n; i++){

key = arr[i];

j = i - 1;

while (j >= 0 && arr[j] > key){

arr[j + 1] = arr[j];

j = j - 1;

}

arr[j + 1] = key;

}

}

int main(){

int n=0;

cout<<"Enter n:";

cin>>n;

int \*arr=new int(n);

arr=Rdm(n);

cout<<"Before Selection Sort:\n";

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

cout<<arr[i]<<” ”;

}

Insertion\_sort(arr,n);

cout<<"After Selection Sort:\n";

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

cout<<arr[i]<<” ”;

}

return 0;

}

## Q5. WAP to implement the quick sort and show the output of each pass.

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//This program is developed by Tanishq Agarwal(211B326)

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Solution:

#include<bits/stdc++.h>

#include"Rdm.h"

using namespace std;

int partition(int arr\*, int start, int end){

int pivot = arr[start];

int count = 0;

for (int i = start + 1; i <= end; i++) {

if (arr[i] <= pivot)

count++;

}

int pivotIndex = start + count;

swap(arr[pivotIndex], arr[start]);

int i = start, j = end;

while (i < pivotIndex && j > pivotIndex) {

while (arr[i] <= pivot) {

i++;

}

while (arr[j] > pivot) {

j--;

}

if (i < pivotIndex && j > pivotIndex) {

swap(arr[i++], arr[j--]);

}

}

return pivotIndex;

}

void quickSort(int\* arr, int start, int end){

if (start >= end)

return;

int p = partition(arr, start, end);

quickSort(arr, start, p - 1);

quickSort(arr, p + 1, end);

}

int main(){

int n=0;

cout<<"Enter n:";

cin>>n;

int \*arr=new int(n);

arr=Rdm(n);

cout<<"Before Selection Sort:\n";

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

cout<<arr[i]<<” ”;

}

quickSort(arr, 0, n - 1);

cout<<"After Selection Sort:\n";

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

cout<<arr[i]<<” ”;

}

return 0;

}

## Q6. WAP to implement the merge sort and show the output of each pass.

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//This program is developed by Tanishq Agarwal(211B326)

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Solution:

#include<bits/stdc++.h>

#include"Rdm.h"

using namespace std;

void merge(int arr[],int l,int m,int r){

int i, j, k=0,0,0;

int n1 = m - l + 1;

int n2 = r - m;

int L[n1], R[n2];

for (i = 0; i < n1; i++){

L[i] = arr[l + i];

}

for (j = 0; j < n2; j++){

R[j] = arr[m + 1 + j];

}

i = 0; j = 0; k = l;

while (i < n1 && j < n2){

if (L[i] <= R[j]){

arr[k] = L[i];

i++;

}

else{

arr[k] = R[j];

j++;

}

k++;

}

while (i < n1) {

arr[k] = L[i];

i++;

k++;

}

while (j < n2){

arr[k] = R[j];

j++;

k++;

}

}

void mergeSort(int arr[],int l,int r){

if (l < r){

int m = l + (r - l) / 2;

mergeSort(arr, l, m);

mergeSort(arr, m + 1, r);

merge(arr, l, m, r);

}

}

int main(){

int n;

cout<<"Enter n:";

cin>>n;

int \*arr=new int(n);

arr=Rdm(n);

cout<<"Before Selection Sort:\n";

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

cout<<arr[i]<<” ”;

}

mergeSort(arr, 0, n - 1);

cout<<"After Selection Sort:\n";

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

cout<<arr[i]<<” ”;

}

return 0;

}

## Advanced Problems:

## Q7. WAP to sort a character array using insertion sort in alphabetic order and print number of shifts.

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//This program is developed by Tanishq Agarwal(211B326)

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Solution:

## Q8. WAP to insert an element in sorted array and after insertion order should not change.

## Sample input : 2, 4, 5, 6, 8, 9, 10, 13, 15 and 7

## Sample output : 2, 4, 5, 6, 7, 8, 9, 10, 13, 15

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//This program is developed by Tanishq Agarwal(211B326)

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Solution:

## Q9. WAP to implement stable selection sort.

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//This program is developed by Tanishq Agarwal(211B326)

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Solution:

## Q10. WAP to implement online insertion sort such that it can sort the numbers entered during the execution of the program

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//This program is developed by Tanishq Agarwal(211B326)

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Solution:

# Lab Exercise 7: Singly Linked List

## Q1. Write a function Insert\_Beginning() to insert a new node at the beginning of

## singly linked list. Call this function N time to create a linked list with N nodes.

## Also write display function to print the linked list.

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//This program is developed by Tanishq Agarwal(211B326)

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Solution:

#include <iostream>

using namespace std;

struct SLLNODE{

int INFO=0;

struct SLLNODE\* NEXT;

};

SLLNODE \*START = NULL;

void insert(int info)

{

SLLNODE \*newnode = new SLLNODE;

newnode->INFO = info;

newnode->NEXT = (START != NULL) ? START : NULL;

START = newnode;

}

void dis()

{

if (START != NULL)

{

cout << "[ " << START->INFO;

SLLNODE \*temp = START->NEXT;

while (temp != NULL)

{

cout << ", " << temp->INFO;

temp = temp->NEXT;

}

cout << " ]\n";

}

else

{

cout << "[ ]\n";

}

}

int main()

{

START = NULL;

cout << "\tCreate linked list using insert\_at\_beg()\n\n"

<< "Enter the size: ";

cin >> n;

for (int i = 0; i < n; i++)

{

int ele;

cout << "Enter " << i << " element: ";

cin >> ele;

insert(ele);

dis();

}

return 0;

}

## Q2. Write a menu driven program using switch-case to insert the node at beginning,

## from specified position and at the end of linked list.

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//This program is developed by Tanishq Agarwal(211B326)

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Solution: #include <iostream>

using namespace std;

struct SLLNODE{

int INFO=0;

struct SLLNODE\* NEXT;

};

SLLNODE \*START = NULL;

void dis()

{

cout << "\nLinked List: [ ";

if (START != NULL)

{

cout << START->INFO;

SLLNODE \*temp = START->NEXT;

while (temp != NULL)

{

cout << ", " << temp->INFO;

temp = temp->NEXT;

}

}

cout << " ]\n";

}

void beg(int info)

{

SLLNODE \*newnode = new SLLNODE;

newnode->INFO = info;

newnode->NEXT = START;

START = newnode;

}

void spec(int info, int loc)

{

SLLNODE \*temp = START, \*newnode = new SLLNODE;

newnode->INFO = info;

if (loc == 1)

{

newnode->NEXT = START;

START = newnode;

}

else if (loc >= 2 && START != NULL)

{

for (int i = 1; i <= loc - 2; i++)

{

temp = temp->NEXT;

if (temp == NULL)

{

cout << "Location out of bounds.\n";

return;

}

}

newnode->NEXT = temp->NEXT;

temp->NEXT = newnode;

}

else

{

cout << "Invalid loc";

}

}

void end(int info)

{

SLLNODE \*newnode = new SLLNODE;

newnode->INFO = info;

newnode->NEXT = NULL;

if (START == NULL)

START = newnode;

else

{

SLLNODE \*temp = START;

while (temp->NEXT != NULL)

temp = temp->NEXT;

temp->NEXT = newnode;

}

}

int main()

{

START = NULL;

int choice, n, loc = -1;

char reload = 'y';

do

{

system("cls||clear");

cout << "\tInserting elements in a linked list\n";

dis();

cout << "Enter the number to be inserted: ";

cin >> n;

cout << "\n1) In the beginning\n";

cout << "2) At a given location\n";

cout << "3) In the end\n";

cout << "\nYour Choice[1 to 3]: ";

cin >> choice;

switch (choice)

{

case 1:

beg(n);

break;

case 2:

cout << "\nEnter the location (start from 1): ";

cin >> loc;

spec(n, loc);

break;

case 3:

end(n);

break;

default:

cout << "Wrong input.";

}

dis();

cout << "\nWould you like to try again? (y): ";

cin >> reload;

} while (reload == 'y');

return 0;

}

## Q3. Write a menu driven program to delete the node from the beginning, from from

## specified position and from the end of linked list.

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//This program is developed by Tanishq Agarwal(211B326)

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Solution: #include <iostream>

using namespace std;

struct SLLNODE{

int INFO=0;

struct SLLNODE\* NEXT;

};

SLLNODE \*START = NULL;

void dis()

{

cout << "\nLinked List: [ ";

if (START != NULL)

{

cout << START->INFO;

SLLNODE \*temp = START->NEXT;

while (temp != NULL)

{

cout << ", " << temp->INFO;

temp = temp->NEXT;

}

}

cout << " ]\n";

}

void beg()

{

if (START != NULL)

{

SLLNODE \*temp = START;

START = START->NEXT;

delete temp;

}

else

{

cout << "Already empty.\n";

}

}

void spec(int loc)

{

SLLNODE \*temp = START, \*temp1 = START;

if (START == NULL || loc <= 0)

cout << "Already empty or Invalid Location.\n";

if (loc == 1)

beg();

else if (loc >= 2 && START->NEXT != NULL)

{

for (int i = 1; i <= loc - 2; i++)

{

temp = temp->NEXT;

if (temp->NEXT == NULL)

cout << "Location out of bounds.\n";

}

temp1 = temp->NEXT;

temp->NEXT = temp1->NEXT;

delete temp1;

}

else

cout << "Location out of bounds.\n";

}

void end()

{

if (START == NULL)

cout << "Already empty.\n";

else if (START->NEXT == NULL)

{

delete START;

START = NULL;

}

else

{

SLLNODE \*temp = START, \*temp1 = START->NEXT;

while (temp1->NEXT != NULL)

{

temp = temp1;

temp1 = temp1->NEXT;

}

temp->NEXT = NULL;

delete temp1;

}

}

int main()

{

START = NULL;

int choice, n, loc = -1;

char reload = 'y';

cout << "\tDeleting elements from a linked list\n\n"

<< "Enter the size: ";

cin >> n;

SLLNODE \*temp = START;

for (int i = 0; i < n; i++)

{

SLLNODE \*newnode = new SLLNODE;

if (i == 0)

START = newnode;

else

temp->NEXT = newnode;

cout << "Enter " << i << " element: ";

cin >> newnode->INFO;

temp = newnode;

}

temp = NULL;

do

{

system("cls||clear");

cout << "\tDeleting elements from a linked list\n";

dis();

cout << "\n1) In the beginning\n";

cout << "2) At a given location\n";

cout << "3) In the end\n";

cout << "\nYour Choice[1 to 3]: ";

cin >> choice;

switch (choice)

{

case 1:

beg();

break;

case 2:

cout << "\nEnter the location (start from 1): ";

cin >> loc;

spec(loc);

break;

case 3:

end();

break;

default:

cout << "Wrong input.";

}

dis();

cout << "\nWould you like to try again? (y): ";

cin >> reload;

} while (reload == 'y');

return 0;

}

## Q4. WAP to reverse the singly linked list.

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//This program is developed by Tanishq Agarwal(211B326)

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Solution: #include <iostream>

using namespace std;

struct SLLNODE{

int INFO=0;

struct SLLNODE\* NEXT;

};

SLLNODE \*START = NULL;

void dis()

{

cout << "\nLinked List: [ ";

if (START != NULL)

{

cout << START->INFO;

SLLNODE \*temp = START->NEXT;

while (temp != NULL)

{

cout << ", " << temp->INFO;

temp = temp->NEXT;

}

}

cout << " ]\n";

}

int main()

{

START = NULL;

int n;

cout << "\tReverse a linked list\n\n"

<< "Enter the size: ";

cin >> n;

SLLNODE \*temp = START;

for (int i = 0; i < n; i++)

{

SLLNODE \*newnode = new SLLNODE;

if (i == 0)

START = newnode;

else

temp->NEXT = newnode;

cout << "Enter " << i << " element: ";

cin >> newnode->INFO;

temp = newnode;

}

temp = NULL;

dis();

cout << "\nReversing the list:";

if (START != NULL || START->NEXT != NULL)

{

temp = START->NEXT;

SLLNODE \*prev = START, \*next = temp->NEXT;

while (temp != NULL)

{

temp->NEXT = prev;

prev = temp;

temp = next;

if (temp == NULL)

break;

next = temp->NEXT;

}

START->NEXT = NULL;

START = prev;

}

dis();

return 0;

}

## Q5. WAP to search an element in the linked list.

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//This program is developed by Tanishq Agarwal(211B326)

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Solution:

#include <iostream>

using namespace std;

struct SLLNODE{

int INFO=0;

struct SLLNODE\* NEXT;

};

SLLNODE \*START = NULL;

void dis()

{

cout << "\nLinked List: [ ";

if (START != NULL)

{

cout << START->INFO;

SLLNODE \*temp = START->NEXT;

while (temp != NULL)

{

cout << ", " << temp->INFO;

temp = temp->NEXT;

}

}

cout << " ]\n";

}

int main()

{

START = NULL;

int n, ele;

bool check;

cout << "\tSearch in a linked list\n\n"

<< "Enter the size: ";

cin >> n;

SLLNODE \*temp = START;

for (int i = 0; i < n; i++)

{

SLLNODE \*newnode = new SLLNODE;

if (i == 0)

START = newnode;

else

temp->NEXT = newnode;

cout << "Enter " << i << " element: ";

cin >> newnode->INFO;

temp = newnode;

}

temp = NULL;

dis();

cout << "\nEnter the element to search: ";

cin >> ele;

temp = START;

int i = 0;

while (temp != NULL)

{

check = (temp->INFO == ele) ? true : false;

if (check)

{

cout << ele << " found at index: " << i << "\n";

break;

}

i++;

temp = temp->NEXT;

}

if (!check)

cout << ele << " not found in the linked list.\n";

return 0;

}

## Q6. Assume you have given a start pointer of a singly linked list. Write a program

## to find the middle node in the given linked list.

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//This program is developed by Tanishq Agarwal(211B326)

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Solution: #include <iostream>

using namespace std;

struct SLLNODE{

int INFO=0;

struct SLLNODE\* NEXT;

};

SLLNODE \*START = NULL;

void dis()

{

cout << "\nLinked List: [ ";

if (START != NULL)

{

cout << START->INFO;

SLLNODE \*temp = START->NEXT;

while (temp != NULL)

{

cout << ", " << temp->INFO;

temp = temp->NEXT;

}

}

cout << " ]\n";

}

int main()

{

START = NULL;

int n;

cout << "\tSearch in a linked list\n\n"

<< "Enter the size: ";

cin >> n;

SLLNODE \*temp = START;

for (int i = 0; i < n; i++)

{

SLLNODE \*newnode = new SLLNODE;

if (i == 0)

START = newnode;

else

temp->NEXT = newnode;

cout << "Enter " << i << " element: ";

cin >> newnode->INFO;

temp = newnode;

}

temp = NULL;

dis();

if (START != NULL)

{

temp = START;

int i = 0;

SLLNODE \*mid = START;

while (temp != NULL && temp->NEXT != NULL)

{

temp = temp->NEXT->NEXT;

mid = mid->NEXT;

i++;

}

cout << "The middle element is: " << mid->INFO << " at index: " << i << "\n";

}

else

{

cout << "The linked list is empty.\n";

}

return 0;

}