OOP LABORATORY 5

Name: **ANIRBAN HAZRA**
Section: **B-12**
Roll : **2005643**

1. WAP to allocate memory for two arrays dynamically. The size of the arrays is given as input. Merge the two arrays into a third array (no sorting required). Deallocate the memory of the first two arrays. Display the elements of the merged (third) array.

PROGRAM CODE:-

```
#include<iostream>
#include<stdlib.h>
using namespace std;

int main ()
{
 int i, j, k, n, m;
        cout << "Enter the number of items in the 1st array:" << "\n";
        cin >> n;
 int *arr1 = new int (n);

 cout << "Enter " << n << " items" << endl;
 for (i = 0; i < n; i++)
  {
       cin >> arr1[i];
  }

 cout << "\nYou entered: ";
 for (i = 0; i < n; i++)
  {
       cout << arr1[i] << " ";
  }

 cout << "\nEnter the number of items in the 2nd array:" << "\n";
 cin >> m;
 int *arr2 = new int (m);

 cout << "\nEnter " << m << " items" << endl;
 for (j = 0; j < m; j++)
  {
       cin >> arr2[j];
  }
```

```
  cout << "\nYou entered: ";
  for (j = 0; j < n; j++)
    {
          cout << arr2[j] << " ";
    }


  int *arr3 = new int (m + n);
  for (i = 0; i < n; i++)
    {
          arr3[i] = arr1[i];
    }

  for (i = 0, j = n; i < m, j < n + m; i++, j++)
    {
          arr3[j] = arr2[i];
    }

    free(arr1); delete[] arr1;
    free(arr2); delete[] arr2;
  cout << "\nThe final merged array is :" ;
    for (i = 0; i < m + n; i++)
    {
          cout << arr3[i] << " ";
    }

  return 0;
}
```

OUTPUT:

```
Enter the number of items in the 1st array:
4
Enter 4 items
2 4 6 8

You entered: 2 4 6 8
Enter the number of items in the 2nd array:
4

Enter 4 items
1 3 5 7

You entered: 1 3 5 7
The final merged array is :2 4 6 8 1 3 5 7
--------------------------------
```

2. WAP to enter a multiline string. Remove all the multiple blank spaces by copying the string to another string. Deallocate the memory for first string. Display the second string.

PROGRAM CODE:-

```
#include<iostream>
#include <stdio.h>
#include <string.h>

using namespace std;

int main(){
   char *x=new char[100], *inputString= new char[100], outputArray[100];
   int readIndex = 0, writeIndex;
   cout<<"Enter a string:";
   cin.getline(x,100,'0');

   for(int i=0;i<strlen(x);i++)
   inputString[i]=x[i];

   delete[] x;

   while(inputString[readIndex] == ' '){
      readIndex++;
   }

   for(writeIndex = 0;inputString[readIndex] != '\0'; readIndex++){
     if(inputString[readIndex]==' ' && inputString[readIndex-1]==' '){
        continue;
     }
     outputArray[writeIndex] = inputString[readIndex];
     writeIndex++;
   }
   outputArray[writeIndex] = '\0';
   cout<<"String without extra spaces\n"<< outputArray;

   return 0;
}
```

OUTPUT:-

```
Enter a string:
My   name  is   Anirban   Hazra
I study   CSE in  KIIT   university.
Odisha0
String without extra spaces

My name is Anirban Hazra
I study CSE in KIIT university.
Odisha
```

3. WAP to enter an integer. Ask the user if he wants to enter another integer. Continue input of integers till user stops. Display all the integers. Use dynamic memory allocation. [Hint: Form link list].

PROGRAM CODE:-

```
#include<iostream>
#include<stdlib.h>
using namespace std;

struct node
{
        int data;
        node *next;
};

class Linked_list
{
        private:
                node *head,*tail;
        public:
           Linked_list()
           {
                head=NULL;
                tail=NULL;
                }

                void add_node(int n)
                {
                        node *tmp=new node;
                        tmp->data=n;
                        tmp->next=NULL;

                        if(head==NULL)
                        {head=tmp;
                          tail=tmp;
                        }
                        else
                        {tail->next=tmp;
                         tail=tail->next;
                        }
                }
                void print()
                {
        if (head == NULL)
        {
                cout<<"List is empty"<<endl;
        }
        else{
                node *tmp = head;
```

```
                cout<<"Linked List: ";
                while (tmp != NULL){
                        cout<<tmp->data<<"->";
                        tmp = tmp->next;
                }
                cout<<"NULL"<<endl;
        }
}
};

int main()
{   int n,p=0;
        Linked_list a;
        m :
                cout<<"Do you want to add an element?\nEnter 1 or 0\n";
                cin>>n;
                if(n==1)
                {
                        cout<<"Enter the element\n";
                        cin>>p;
                        a.add_node(p);
                        p=0;
                        goto m;
            }
            else cout<<"\n Your list is :";
            a.print();
}
```

OUTPUT:-

```
Do you want to add an element?
Enter 1 or 0
1
Enter the element
4
Do you want to add an element?
Enter 1 or 0
1
Enter the element
6
Do you want to add an element?
Enter 1 or 0
1
Enter the element
8
Do you want to add an element?
Enter 1 or 0
0

 Your list is :Linked List: 4->6->8->NULL
```