

WEEK – 2

CODE HACKATHON

(Department-CSE)

Date: 4th August 2018

Total Marks : 100

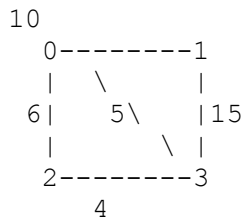
Created by: - Nishanth ¾ B-16 - 7093197160

V.Srinivas ¾ B-16 - 8374256375

Ajit panigrahi ¾ B-12- 8249852901

M.pranay ¾ B-16 - 8919345427

1) Write the program to find the minimum spanning tree for the following graph: (15-MARKS)



Sample output:

Following are the edges in the constructed MST

```
2 -- 3 == 4
0 -- 3 == 5
0 -- 1 == 10
```

Q2) Speak like Yoda from Star Wars, where "I am Pranay" becomes, "Pranay, I am."

[For the input line, shift the last word to the first and add a comma & space.]

(10-MARKS)

Input:

We play monopoly.

Output:

Monopoly, we play.

[Marks for: capitalize monopoly, lowercase we, fullstop placement]

Q3) . You are given a function - Node* mergelists(Node* head1,Node* head2).This function takes two head pointers of two different SORTED Linked lists.(10-MARKS)

Given these two head pointers, MANIPULATE THE LINKS between the two SORTED lists such that it forms a Merged List that is still SORTED.

Return the head pointer that points to the third SORTED list after manipulating the list via the function "mergelists".

YOU ONLY HAVE TO RETURN THE POINTER , THE OUTPUT IS ALREADY MANAGED.

COMMENT OUT THE FUNCTION "mergelists" AND COMPLETE IT ACCORDING TO THE TASK.

SAMPLE CASE:

List 1:

1->2->3->NULL

List 2:

3->4->NULL

OUTPUT LIST:

1->2->3->3->4->NULL

OUTPUT LIST STRUCTURE:

1->2->3-|

|->3->4->NULL.

IMPORTANT NOTE:Either of the head pointer passed to the function head1 or head2 may be NULL.

```
*/
```

```
#include<iostream>
```

```
using namespace std;
```

```
class Node
```

```
{
```

```
public:
```

```
    int data;
```

```
    Node* next;
```

```
    Node(int n)
```

```
{
```

```
    this->data=n;
```

```
    this->next=NULL;
```

```
}
```

```
};
```

```
class List
```

```
{
```

```
public:
```

```
    Node* head;
```

```
    Node* tail;
```

```
    List()
```

```

{
    this->head=NULL;
    this->tail=NULL;
}

void insert_node(int data)
{
    Node* n=new Node(data);
    if(this->head==NULL)
    {
        this->head=n;
    }
    else
    {
        this->tail->next=n;
    }
    this->tail=n;
}

void display_list()
{
    if(head==NULL)
        cout<<"EMPTY"<<endl;
    Node* temp=head;
    while(temp)
    {
        cout<<temp->data<<" ";
        temp=temp->next;
    }
    cout<<endl;
}

};

```

```

Node* mergelists(Node* head1,Node *head2)//Complete this
function.....

// such that it return a pointer t
the merge list.

{
}

//.....
.

main()
{

    int n1,n2;
    int value;
    cout<<"Enter the number of nodes in the first list:";
    cin>>n1;
    cout<<"Enter the number of nodes in the second list:";
    cin>>n2;
    List* l1=new List();
    List* l2=new List();
    List* l3=new List();
    //Input List 1.....
    if(n1>0)
    {
        cout<<"Enter the first list:";
        for(int i=0;i<n1;i++)
        {
            cin>>value;
            l1->insert_node(value);
        }
    }
}

```

```

//Input List 2.....
if(n2>0)
{
    cout<<"Enter the second list:";
    for(int i=0;i<n2;i++)
    {
        cin>>value;
        l2->insert_node(value);
    }
}
//.....

//Displaying the Initial Lists.....
cout<<"List 1:"<<endl;
l1->display_list();
cout<<endl;
cout<<"List 2:"<<endl;
l2->display_list();
cout<<endl;

//Output.....
cout<<"Output:"<<endl;
l3->head=mergelists(l1->head,l2->head);
l3->display_list();
//.....
}

```

Q4) C Example to use 'kbhit' function
(10-MARKS)

Sample output: "Press a key" will keep printing on the

Console until the user presses a key on the keyboard.

Q5) There are N couples in cinema hall and one more person who came alone. For given $2*N+1$ input elements, find that unique individual if all of the couples are denoted by the same number.

(5-MARKS)

Input:

88 120 34 2 345 3 2 90 88 120 3 90 34

Output:

345

Q6) you are given two Cars on a number line ready to move in the positive direction (i.e, toward positive infinity).

-The first Car starts at location x_1 and moves at a rate of v_1 meters per unit time.

-The second Car starts at location x_2 and moves at a rate of v_2 meters per time.

You have to figure if we can get both Cars at the same location at the same time. If it is possible, return "YES", otherwise return "NO".

(10-MARKS)

INPUT:

A single line of four space-separated integers denoting the respective values of x_1, v_1, x_2, v_2 .

OUTPUT:

YES if they will meet.

NO if they won't.

Example:

Car1:

(Position,Velocity)=(0,3)

Car2:

(Position,Velocity)=(4,2)

After 4 Seconds:

Position of Car1= $0+(4*3)=12$

Position of Car2= $4+(4*2)=12$

They meet therefore we print Yes.

Q7) Program to print weekday of given date (10 MARKS)

Sample output:

First Run:

```
Input date (DD-MM-YYYY): 22-02-2015
Date is correct [22/02/2015].
Week day is: Sunday
```

Second Run:

```
Input date (DD-MM-YYYY): 01-01-2016
Date is correct [01/01/2016].
Week day is: Friday
```

Third Run:

```
Input date (DD-MM-YYYY): 30-02-2010
Date is in-correct.
```

Fourth Run:

```
Input date (DD-MM-YYYY): 29-02-2001
Date is in-correct.
```

Fifth Run:

```
Input date (DD-MM-YYYY): 30-13-2011
Date is in-correct.
```

Q8) Nishant plays with a list of numbers and decides to cycle its elements in alternate directions in the fibonacci series. First he shifts every element one step forward, the last element moved to the first, then he shifts every element one step backwards, moving the first element to the back. Then he shifts elements by 2 positions forward, and then 3 positions back. For a given string, produce the final string after Nth shift in the fibonacci series. (15 MARKS)

Input:

```
Nishant
5
```

Output:
hantNis

```
[+1, -1, +2, -3, +5]
```

Q9) C program to find Binary Addition and Binary Subtraction (take the numbers in decimal format and print them in decimal format) (5-MARKS)

Sample output:

```
Input first integer value: 30
Input second integer value: 5
```


Binary Addition: 35
Binary Subtraction: 25

Q10) A palindrome is a word, phrase, number, or other sequence of characters which reads the same backwards and forwards. Determine if a given string - S is a palindrome.

To solve this challenge, we must first take each character in , enqueue it in a queue, and also push that same character onto a stack. Once that's done, we must dequeue the first character from the queue and pop the top character off the stack, then compare the two characters to see if they are the same; As long as the characters match, we continue dequeuing, popping, and comparing each character until our containers are empty (a non-match means isn't a palindrome).

Complete the methods below as asked. (10-MARKS)

SAMPLE INPUT:
aaa

SAMPLE OUTPUT:
PALINDROME

```
*/
#include <iostream>
#include<stdlib.h>
#include<string.h>
using namespace std;
char a[100];//A Stack represented by a Global Array.
char b[100];//A Queue represented by a Global Array.
int i=0,j=0,k=0;//Three variables index the Data structures. You may use
them as you like or define your own variables instead of these.

void pushCharacter(char ch)//Adds a character to the Stack.
{
    //Complete this function..
}
void enqueueCharacter(char ch)//Adds a character to Queue.
{
    //Complete this function..
}
char popCharachter()
{
    //Complete this function..
}

char dequeCharachter()
{
    //Complete this function.
}
main(){
    string str;
    cout<<"Enter string:";
```

```
cin>>str;
for(int i=0;i<str.length();i++)
{
    pushCharacter(str[i]);
    enqueueCharacter(str[i]);
}
bool same=true;
for(int i=str.length()-1;i>=0;i--)
{
    same=same&&(popCharacter()==dequeCharacter());
}
if(same)
    cout<<"PALINDROME"<<endl;
else
    cout<<"NOT PALNDROME"<<endl;
}
```

-----THE END-----

ALL THE BEST ☺