**UE20CS207 - DATA STRUCTURES AND ITS APPLICATIONS LABORATORY**

**30th August to 4th September 2021**

|  |  |  |  |
| --- | --- | --- | --- |
| Vanshika Goel | **PES1UG20CS484** | H Section | Roll No: 40 |

**RECURSION(5 + 5 MARKS)**

(i) Write a recursive function to search for a substring within a given string. Return True if substring is found, else return false  
Ex: input :  
string : This is our first DS Lab

substring : DS output : True

string : This is our first DS Lab substring : DBMS  
output : False

Input:

Text

Description automatically generated

Output:

Text

Description automatically generated

(ii) Write a recursive function to generate all permutations of a given text Ex : input : ABC, output : ABC,ACB, BAC, BCA, CAB, CBA

Text

Description automatically generated

Output:

Text

Description automatically generated

**STRUCTURE AND DYNAMIC MEMORY ALLOCATION ( 10 MARKS)**

1). Write a C program to create a record for few students containing the student details as SRN, Name, semester, marks for 5 subjects. Each subject should have a code associated with it  
a. Compute the class average marks in a particular subject.  
b. Sort the students based on SRN.

Input:

Text

Description automatically generated

Text

Description automatically generated

Output:

Text

Description automatically generated with medium confidence

**ASSIGNMENT : (10 MARKS)**

1. Perform Tower of Hanoi using recursion

Input:

#include <stdio.h>

void toh(int, char, char, char);

int main()

{

int n;

printf("Enter no of disks: ");

scanf("%d", &n);

printf("For Tower of Hanoi: \n");

toh(n, 'A', 'C', 'B');

return 0;

}

void toh(int n, char from, char to, char aux)

{

if (n == 1)

{

printf("Move disk 1 from %c to %c \n", from, to);

return;

}

toh(n - 1, from, aux, to);

printf("Move disk %d from %c to %c \n", n, from, to);

toh(n - 1, aux, to, from);

}

Output:

Text

Description automatically generated

**2.** Write a C program to create a record for IPL Players containing the details as player name, team name, no of matches played, runs scored.  
a. Display the player details who scored maximum runs in a particular match.  
b. Also compute the number of matches played by each player.

Input:

#include<stdio.h>

#include<string.h>

#include<stdlib.h>

struct match

{

char played[15];

int runs[14];

};

typedef struct match MATCH;

struct ipl

{

char player\_name[50];

char team\_name[50];

MATCH dets;

};

typedef struct ipl IPL;

int main()

{

int c='y';

int i=0;

IPL \*s=(IPL\*)malloc(sizeof(IPL));

while(c=='y')

{

printf("Enter the name of player: \n");

scanf("%s",(s+i)->player\_name);

fflush(stdin);

printf("Enter the name of team: \n");

scanf("%s",(s+i)->team\_name);

fflush(stdin);

for(int j=0;j<10;j++)

{

printf("Did the player play in match %d? \n",j+1);

printf("Enter y/n: \n");

scanf("%c",&((s+i)->dets.played[j]));

fflush(stdin);

if((s+i)->dets.played[j]=='y')

{

printf("Enter the runs scored by the player: \n");

scanf("%d",&((s+i)->dets.runs[j]));

fflush(stdin);

}

else

(s+i)->dets.runs[j]=-1;

}

printf("Do you wish to enter more data: \n");

printf("Enter y/n: \n");

c=getchar();

fflush(stdin);

if(c=='y')

{

i++;

s=(IPL\*)realloc(s,sizeof(IPL)\*(i+1));

}

}

int max=-1;

int match\_n;

printf("Enter the match number for calculation of max: \n");

scanf("%d",&match\_n);

match\_n--;

int index=-1;

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

{

if((s+j)->dets.runs[match\_n]>max)

{

max=(s+j)->dets.runs[match\_n];

index=j;

}

}

if(max==-1)

{

printf("No player played that match. \n");

}

else

{

printf("The player is %s who scored %d in match %d. \n",(s+index)->player\_name,max,++match\_n);

}

int count=0;

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

{

count=0;

for(int k=0;k<14;k++)

{

if((s+j)->dets.played[k]=='y')

count++;

}

printf("Player %s played %d matches. \n",(s+j)->player\_name,count);

}

return 0;

}

Output:

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generatedText

Description automatically generatedText

Description automatically generated