WEEK-1

PROGRAM-1

Q. Check if a given sub string is present in a given main string :

```
#include<string.h>

#include<stdio.h>

#include<ctype.h>

int findSubString(char* mainString,char* subString,int mainLength,int subLength,int position,int subPosition)

{


	if(position >= mainLength || subPosition >= subLength)

	{

		return 1;

	}




	if(isspace(subString[subPosition]))

	{

		return 1;

	}




	if(mainString[position] == subString[subPosition])

	{

		return findSubString(mainString,subString,mainLength,subLength,position + 1,subPosition + 1);

	}




	return 0;
```

```
}



int main()
{

        char mainString[100];
        char subString[100];


        int mainLength;
        int subLength;
        int i;


        int found = 0;


        printf("Enter main string : ");
        fgets(mainString, 100, stdin);


        printf("Enter sub string : ");
        fgets(subString, 100, stdin);


        mainLength = strlen(mainString);
        subLength = strlen(subString);
```

```
	for(i = 0; i < mainLength - 1 ; i++)
	{
		if(mainString[i] ==  subString[0])
		{

			if(findSubString(mainString,subString,mainLength,subLength,i,0))
			{
				printf("Input substring is found on position %d ",i+1);
				found = 1;
				break;
			}
		}
	}


	if(found == 0)
	{
		printf("Input substring is not found in the string...");
	}
}
```

PROGRAM-2

Q. Print all the permutations of the given string :

```
#include<stdio.h>
#include<string.h>
void swap(char *x, char *y)
{
    char temp;
    temp = *x;
    *x = *y;
    *y = temp;
}

void permute(char *a, int l, int r)
{
int i;
if (l == r)
```

```
    printf("%s\n", a);
else
{
    for (i = l; i <= r; i++)
    {
        swap((a+l), (a+i));
        permute(a, l+1, r);
        swap((a+l), (a+i));
    }
}
}

int main()
{
    char str[10];
            printf("Enter the string : \n");
            scanf("%s", str);
    int n = strlen(str);
    permute(str, 0, n-1);
    return 0;
}
```

PROGRAM-3

Q. Record containing SRN, name, semester and marks of 5 subjects of few students :

```
#include<stdio.h>
#include<string.h>
typedef struct stud
{
char SRN[14];
char name[25];
int sem;
struct marks
{
int PH101;
int MA151;
int CS151;
int ME101;
int EE101;
}m;
```

```
}stud;
void swap(stud*,stud*);
void main()
{
int n;
printf("Enter number of students: ");
scanf("%d",&n);
stud s[n+1];
for(int i=0;i<n;i++)
{
printf("\nSRN: ");
scanf(" %[^\n]s",s[i].SRN);
printf("Name: ");
scanf(" %[^\n]s",s[i].name);
printf("Sem: ");
scanf("%d",&s[i].sem);
printf("Marks for PH101, MA151, CS151, ME101, EE101:\n");
scanf("%d%d%d%d
%d",&s[i].m.PH101,&s[i].m.MA151,&s[i].m.CS151,&s[i].m.ME101,&s[i].m.EE101);
}
float PHsum=0,MAsum=0,CSsum=0,MEsum=0,EEsum=0;
//a)
for(int i=0;i<n;i++)
{
PHsum+=s[i].m.PH101;
MAsum+=s[i].m.MA151;
CSsum+=s[i].m.CS151;
MEsum+=s[i].m.ME101;
EEsum+=s[i].m.EE101;
}
```

```
printf("\n\nClass average marks in:\nPhysics(PH101)=%.2f\nMaths(MA151)=%.2f\nComputer Science(CS151)=%.2f\nMechanics(ME101)=%.2f\nElectrical(EE101)= %.2f\n\n",PHsum/n,MAsum/n,CSsum/n,MEsum/n,EEsum/n);

        //b)

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

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

        if(strcmp(s[j].SRN,s[j+1].SRN)>=1)

        swap(&s[j],&s[j+1]);

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

        {

        printf("\nSRN: %s",s[i].SRN);

printf("\nName: %s",s[i].name);

printf("\nSem: %d",s[i].sem);

printf("\nMarks:\nPH101:%d\nMA151:%d\nCS151:%d\nME101:%d\nEE101:%d\n",s[i].m.PH101, s[i].m.MA151,s[i].m.CS151,s[i].m.ME101,s[i].m.EE101);

}

}

void swap(stud* n1, stud* n2)

{

stud temp=*n1;

*n1=*n2;

*n2=temp;

}
```


Command Prompt
C:\Users\Sanam\Desktop>gcc record.c
C:\Users\Sanam\Desktop>a
Enter number of students: 3
SRN: 1
Name: s
Sem: 3
Marks for PH101, MA151, CS151, ME101, EE101:
5
6
7
8
9
SRN: 2
Name: d
Sem: 3
Marks for PH101, MA151, CS151, ME101, EE101:
3
6
9
6
8
SRN: 3
Name: e
Sem: 3
Marks for PH101, MA151, CS151, ME101, EE101:
4
8
6
8
6
Class average marks in:
Physics(PH101)=4.00
Maths(MA151)=6.67
Computer Science(CS151)=7.33
Mechanics(ME101)=7.33
Electrical(EE101)=7.67
SRN: 1
Name: s
Sem: 3
Marks:
PH101:5


Command Prompt
6
Class average marks in:
Physics(PH101)=4.00
Maths(MA151)=6.67
Computer Science(CS151)=7.33
Mechanics(ME101)=7.33
Electrical(EE101)=7.67
SRN: 1
Name: s
Sem: 3
Marks:
PH101:5
MA151:6
CS151:7
ME101:8
EE101:9
SRN: 2
Name: d
Sem: 3
Marks:
PH101:3
MA151:6
CS151:9
ME101:6
EE101:8
SRN: 3
Name: e
Sem: 3
Marks:
PH101:4
MA151:8
CS151:6
ME101:8
EE101:6
C:\Users\Sanam\Desktop>

WEEK-5

PROGRAM

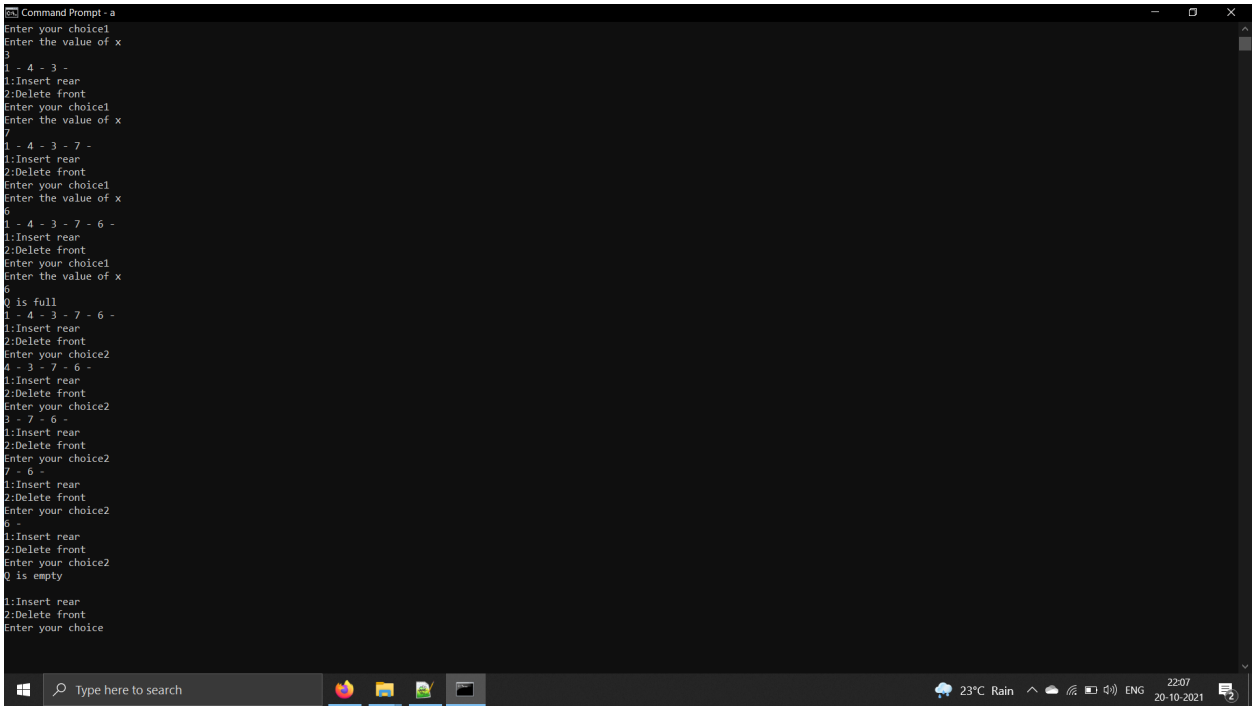
Q. Queue with proper limits :

```
#include<stdio.h>
#include<stdlib.h>
int insert_rear(int *q, int *f, int *r, int size, int x);
int delete_front(int *q, int *f, int *r, int size);
void display(int *q, int f, int r, int size);
int main()
{
	int *q;
	int f, r, size, ch, k, x;
	f=-1;
	r=-1;
	printf("Enter the size of Q\n");
	scanf("%d", &size);
	q=(int *)malloc(size*(sizeof(int)));
	while(1)
	{
		display(q, f, r, size);
		printf("\n1:Insert rear\n");
		printf("2:Delete front\n");
		printf("Enter your choice");
		scanf("%d", &ch);
		switch(ch)
		{
			case 1:printf("Enter the value of x\n");
				scanf("%d", &x);
				insert_rear(q, &f, &r, size, x);
```

```
                    break;
                case 2:delete_front(q, &f, &r, size);
                    break;
                case 3:exit(0);
            }
        }
}

int insert_rear(int *q, int *f, int *r, int size, int x)
{
    if(*r==size-1)
    {
        printf("Q is full\n");
        return -1;
    }
    (*r)++;
    q[*r]=x;
        if(*f==-1)
            *f=0;
    return 1;
}

int delete_front(int *q, int *f, int *r, int size)
{
        int x;
        if(*f==-1)
        {
            printf("Q is empty\n");
            return -1;
        }
```

```
        x=q[*f];
        if(*f==*r)
            *f=*r=-1;
        else
           (*f)++;
        return x;
}

void display(int *q, int f, int r, int size)
{
    int i;
    if(f==-1)
        printf("Q is empty\n");
    else
    {
        for(i=f; i<=r; i++)
        {
                printf("%d - ", q[i]);
        }
    }
}
```


Command Prompt - a
C:\Users\Sanam\Desktop>gcc queue.c
C:\Users\Sanam\Desktop>a
Enter the size of Q
5
Q is empty
1:Insert rear
2:Delete front
Enter your choice1
Enter the value of x
1
1 -
1:Insert rear
2:Delete front
Enter your choice1
Enter the value of x
4
1 - 4 -
1:Insert rear
2:Delete front
Enter your choice1
Enter the value of x
3
1 - 4 - 3 -
1:Insert rear
2:Delete front
Enter your choice1
Enter the value of x
7
1 - 4 - 3 - 7 -
1:Insert rear
2:Delete front
Enter your choice1
Enter the value of x
6
1 - 4 - 3 - 7 - 6 -
1:Insert rear
2:Delete front
Enter your choice1
Enter the value of x
6
Q is full
1 - 4 - 3 - 7 - 6 -
1:Insert rear
2:Delete front
Enter your choice2
4 - 3 - 7 - 6 -
1:Insert rear


Command Prompt - a
Enter your choice1
Enter the value of x
3
1 - 4 - 3 -
1:Insert rear
2:Delete front
Enter your choice1
Enter the value of x
7
1 - 4 - 3 - 7 -
1:Insert rear
2:Delete front
Enter your choice1
Enter the value of x
6
1 - 4 - 3 - 7 - 6 -
1:Insert rear
2:Delete front
Enter your choice1
Enter the value of x
6
Q is full
1 - 4 - 3 - 7 - 6 -
1:Insert rear
2:Delete front
Enter your choice2
4 - 3 - 7 - 6 -
1:Insert rear
2:Delete front
Enter your choice2
3 - 7 - 6 -
1:Insert rear
2:Delete front
Enter your choice2
7 - 6 -
1:Insert rear
2:Delete front
Enter your choice2
6 -
1:Insert rear
2:Delete front
Enter your choice2
Q is empty
1:Insert rear
2:Delete front
Enter your choice
Type here to search
23°C Rain
ENG
22:07
20-10-2021

WEEK-6

PROGRAM-1

Q. Josephus problem :

```
#include<stdio.h>
#include<stdlib.h>
typedef struct node
{
    int data;
    struct node* link;

}NODE;

typedef struct queue
{
    NODE* front;
    NODE* rear;
}Queue;



void init(Queue* pq)
{
    pq->front = NULL;
    pq->rear = NULL;
}



void enqueue(Queue *pq,int, int, int);
int dequeue_skip(Queue *pq, int);
```

```
int main()
{
    Queue pq;
    init(&pq);
    int n,skip,i;
    printf("enter the no. of person\n");
    scanf("%d", &n);
    printf("enter the number of skips\n");
    scanf("%d", &skip);

    for(i = 1;i<=n;i++)
        enqueue(&pq,i,n,skip);
    int winner = dequeue_skip(&pq, skip);
    printf("Survivor is %d\n",winner );

return 0;
}


void enqueue(Queue *pq, int value, int n, int skip)
{

    NODE* temp = (NODE*)malloc(sizeof(NODE));
    temp->data = value;
    temp->link = NULL;

    if (pq->front == NULL && pq->rear == NULL)
    {
        pq->front = temp;
        pq->rear = temp;
```

```
        pq->rear->link = pq->front;
    }

    else
    {
        pq->rear->link = temp;
        pq->rear = temp;
        temp->link = pq->front;
    }

}

int dequeue_skip(Queue *pq, int skip)
{

    int i;
    NODE *p, *befp;
    p = pq->front;
    befp = NULL;
    while(p->link!=p)
    {
        for(i = 0;i<skip-1;i++)
        {
            befp = p;
            p = p->link;
        }
        befp->link = p->link;


             p = befp->link;
```

```
    }
    pq->front = p;
    return (p->data);
}
```

PROGRAM-2

Q. Creat a queue using 2 stacks :

```
#include <stdio.h>
#include <stdlib.h>

void push1(int);
void push2(int);
int pop1();
int pop2();
void enqueue();
void dequeue();
void display();
```

```
int st1[100], st2[100];
int top1 = -1, top2 = -1;
int count = 0;

int main()
{
    int ch;

    printf("\n1 - Enqueue element into queue");
    printf("\n2 - Dequeu element from queue");
    printf("\n4 - Exit\n");

    while (1)
    {
        display();
        printf("\nEnter choice : ");
        scanf("%d", &ch);
        switch (ch)
        {
        case 1:
            enqueue();
            break;
        case 2:
            dequeue();
            break;
        case 4:
            exit(0);
        }
```

```
    }
}

void push1(int data)
{
    st1[++top1] = data;
}

int pop1()
{
    return(st1[top1--]);
}

void push2(int data)
{
    st2[++top2] = data;
}

int pop2()
{
    return(st2[top2--]);
}

void enqueue()
{
    int data, i;

    printf("Enter data into queue : ");
    scanf("%d", &data);
    push1(data);
```

```
    count++;
}

void dequeue()
{
    int i;

    for (i = 0;i <= count;i++)
    {
        push2(pop1());
    }
    pop2();
    count--;
    for (i = 0;i <= count;i++)
    {
        push1(pop2());
    }
}

void display()
{
    int i;
    if(top1<0)
      {
            printf("Queue is empty\n");
      }
    else
    {
      for (i = 0;i <= top1;i++)
      {
```

```
            printf(" %d ", st1[i]);
        }
    }
}
```

WEEK-7

PROGRAM-1

Q. Creat Binary Search tree to store SRNs :

```
#include<stdio.h>
#include<stdlib.h>
void insert(char *t, char *SRN);
void sort(char *t, int i);
void search(char *t, char *SRN);
int main()
{
        int ch, i;
        char t[100];
        char SRN[10];
        for(i=0;i<100;i++)
                t[i]=-1;
        while(1)
        {
                printf("\n1:Insert\n2:Sort\n3:Search\n");
                printf("Enter your choice : \n");
                scanf("%d", &ch);
                switch(ch)
                {
                        case 1:printf("Enter the SRN : \n");
                                scanf("%s", SRN);
                                insert(t, SRN);
                                break;
                        case 2:sort(t, 0);
                                break;
```

```
                        case 3:printf("Enter the SRN you want : \n");
                                scanf("%s", SRN);
                                search(t, SRN);
                                break;
                }

        }
}

void insert(char *t, char *SRN)
{
        int i=0;
        while(t[i]!=-1)
        {
                if(*SRN<t[i])
                        i=2*i+1;
                else
                        i=2*i+2;
        }
        t[i]=*SRN;
}

void sort(char *t, int i)
{
        if(t[i]!=-1)
        {
                sort(t, 2*i+1);
                printf("%c ", t[i]);
                sort(t, 2*i+2);
        }
```

```
}

void search(char *t, char *SRN)
{
        int i=0;
        while(t[i]!=*SRN)
                i++;
        printf("SRN = %c\n", *SRN);
        printf("Position = %d\n", i);      }
```

PROGRAM-2

Q. Creat a Binary tree and check if it's a Binary Search tree :

```
#include<stdio.h>
#include<stdlib.h>
void insert(int *t, int key);
void preorder(int *t, int i);
int ifBTS(int *t, int i);
int main()
{
        int i, ch, num;
        int t[100];
        for(i=0;i<100;i++)
         t[i]=-1;
        while(1)
 {
         printf("\n1:Insert\n2:Preorder\n3:Check for Binary search tree\n");
         printf("Enter your choice : \n");
         scanf("%d", &ch);
         switch(ch)
```

```
        {
                case 1:printf("Enter the value : ");
                       scanf("%d", &num);
                       insert(t, num);
                       break;
                case 2:preorder(t,0);
                       break;
                case 3:if(ifBTS(t,0))
                {
                                printf("The above tree is a Binary search tree\n");
                }
                else
                {
                                printf("The given tree is not a Binary search tree\n");
                }
                                break;
        }
 }
}

void insert(int *t, int key)
{
        int i=0;
        while(t[i]!=-1)
         i++;
        t[i]=key;
}

void preorder(int *t, int i)
{
```

```
        if(t[i]!=-1)
        {
                printf("%d ", t[i]);
                preorder(t, 2*i+1);
                preorder(t, 2*i+2);
        }
}

int ifBTS(int *t, int i)
{
        if(t[i]!=-1)
        {
                if(t[i]<t[2*i+2] && t[i]>t[2*i+1])
                {
                        ifBTS(t, 2*i+1);
                        ifBTS(t, 2*i+2);
                        return 1;
                }
                else
                        return 0;
 }

}
```


Command Prompt - a
C:\Users\Sanam\Desktop\week1>a
1:Insert
2:Preorder
3:Check for Binary search tree
Enter your choice :
1
Enter the value : 1
1:Insert
2:Preorder
3:Check for Binary search tree
Enter your choice :
1
Enter the value : 2
1:Insert
2:Preorder
3:Check for Binary search tree
Enter your choice :
1
Enter the value : 3
1:Insert
2:Preorder
3:Check for Binary search tree
Enter your choice :
2
1 2 3
1:Insert
2:Preorder
3:Check for Binary search tree
Enter your choice :
3
The given tree is not a Binary search tree
1:Insert
2:Preorder
3:Check for Binary search tree
Enter your choice :


Command Prompt - a
C:\Users\Sanam\Desktop\week1>gcc 2.c
C:\Users\Sanam\Desktop\week1>a
1:Insert
2:Preorder
3:Check for Binary search tree
Enter your choice :
1
Enter the value : 4
1:Insert
2:Preorder
3:Check for Binary search tree
Enter your choice :
1
Enter the value : 3
1:Insert
2:Preorder
3:Check for Binary search tree
Enter your choice :
1
Enter the value : 5
1:Insert
2:Preorder
3:Check for Binary search tree
Enter your choice :
1
Enter the value : 2
1:Insert
2:Preorder
3:Check for Binary search tree
Enter your choice :
2
4 3 2 5
1:Insert
2:Preorder
3:Check for Binary search tree
Enter your choice :
3
The above tree is a Binary search tree
1:Insert
2:Preorder
3:Check for Binary search tree
Enter your choice :