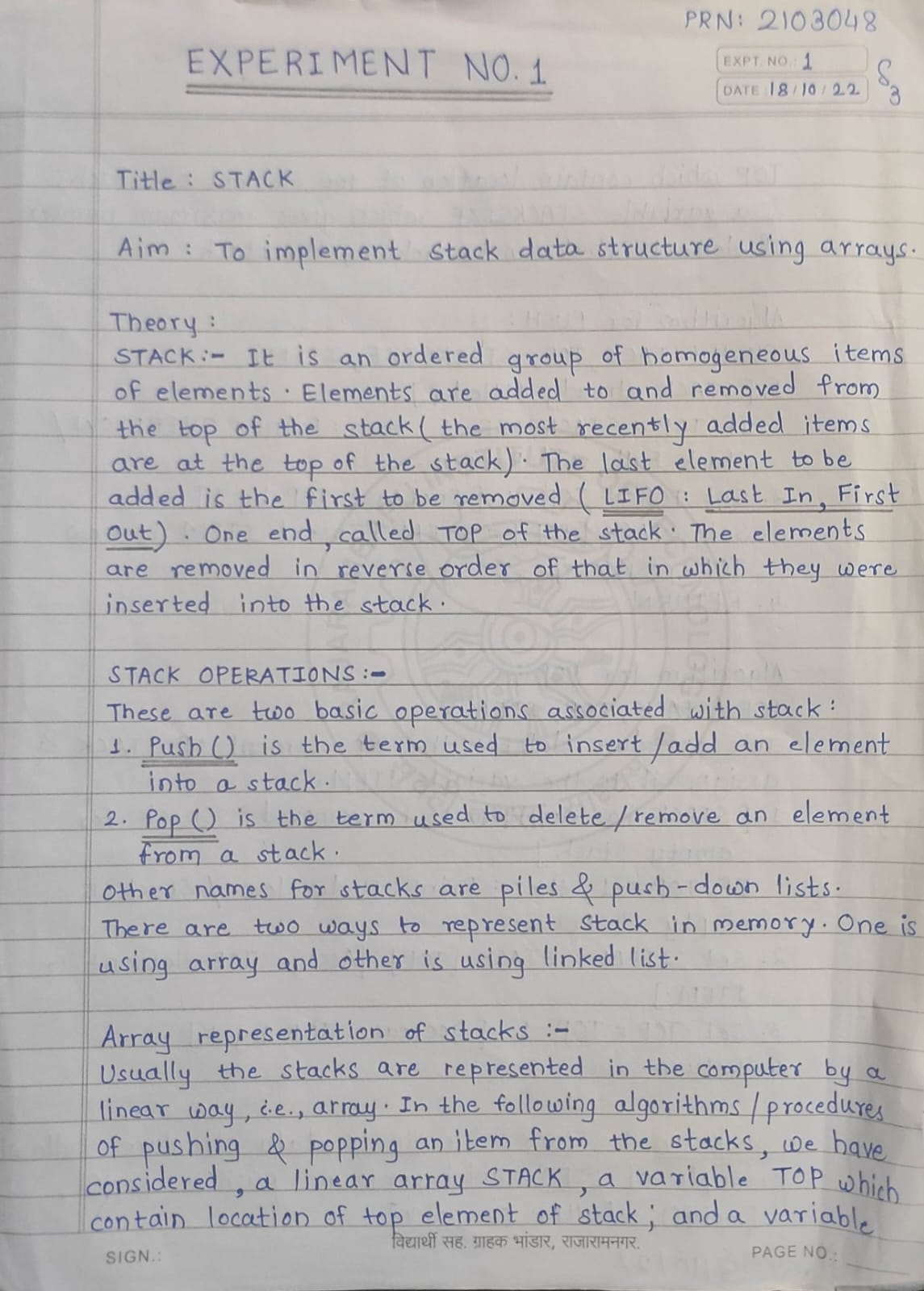
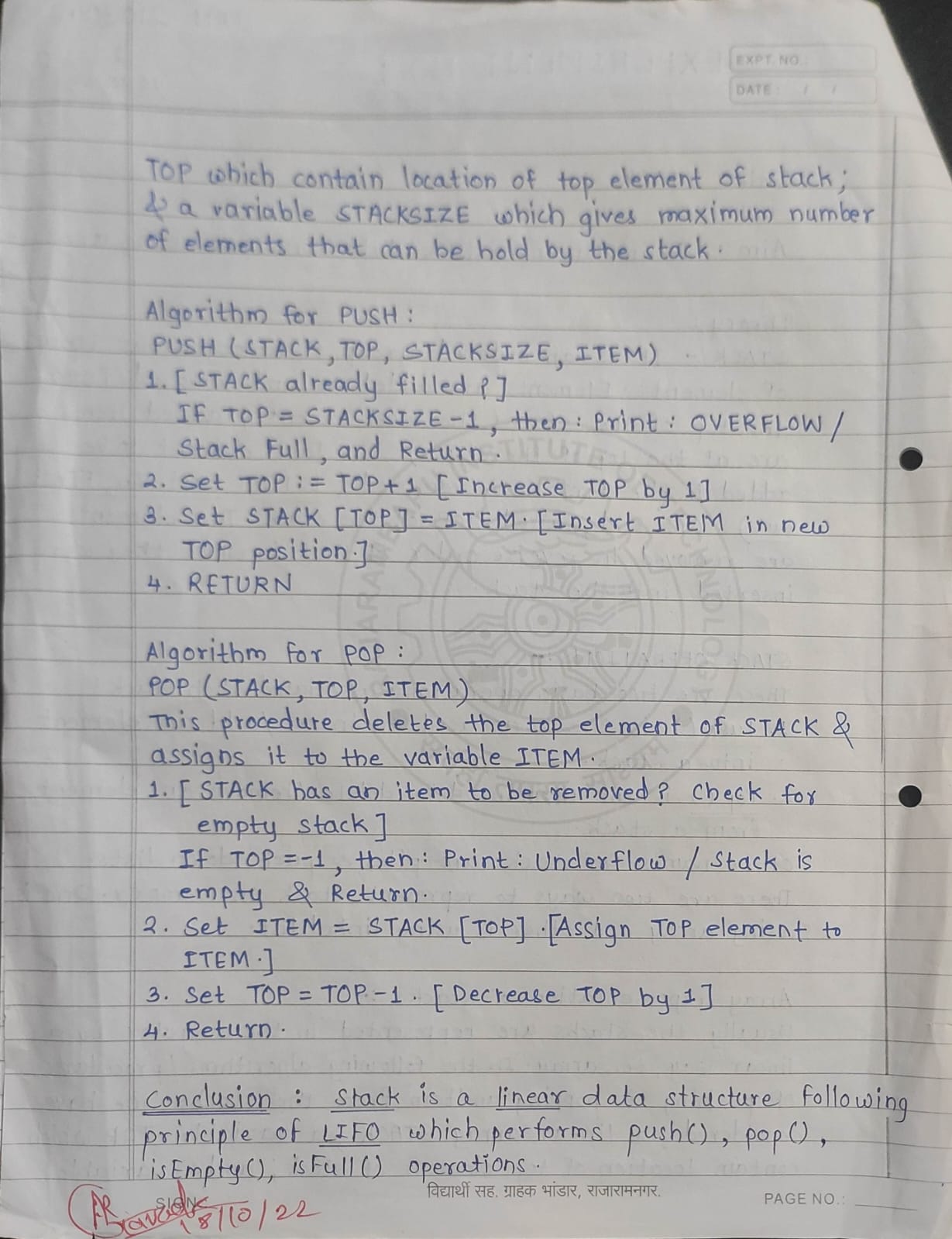
# 2103048

EXPERIMENT 1

FF

1. STACK USING ARRAY :-

CODE :

#include<stdio.h>

int stack[100],ch,n,top,i,x;

void push(); // nA,nR

void pop();

void display();

int main()

{

top=-1;

printf("Enter size of stack:\n");

scanf("%d",&n);

printf("1.Push\t2.Pop\t3.Display\t4.Exit\n");

do

{

printf("Enter your choice:\n");

scanf("%d",&ch);

switch(ch)

{

case 1:

push();

break;

case 2:

pop();

break;

case 3:

display();

break;

case 4:

printf("Exit point!");

break;

default:

printf("Enter valid choice(1/2/3/4)\n");

}

}

while(ch!=4);

return 0;

}

void push()

{

if(top>=n-1)

printf("Stack is OVERFLOW\n");

else

{

printf("Enter a value to be added\n");

scanf("%d",&x);

top++;

stack[top]=x;

}

}

void pop()

{

if(top<=-1)

printf("Stack is UNDERFLOW\n");

else

{

printf("%d is a value to be removed\n",stack[top]);

top--;

}

}

void display()

{

if(top>-1)

{

printf("The elements in stack:\n");

for(i=top;i>=0;i--)

printf("%d\n",stack[i]);

printf("Enter next choice:\n");

}

else

printf("Stack is EMPTY\n");

}

OUTPUT :

Enter size of stack:

3

1.Push 2.Pop 3.Display 4.Exit

Enter your choice:

1

Enter a value to be added

22

Enter your choice:

1

Enter a value to be added

33

Enter your choice:

1

Enter a value to be added

44

Enter your choice:

1

Stack is OVERFLOW

Enter your choice:

2

44 is a value to be removed

Enter your choice:

3

The elements in stack:

33

22

Enter next choice:

Enter your choice:

2

33 is a value to be removed

Enter your choice:

2

22 is a value to be removed

Enter your choice:

2

Stack is UNDERFLOW

Enter your choice:

4

Exit point!

1. REVERSE STRING USING STACK :-

CODE :

#include<stdio.h>

#include<string.h>

int t,s[100];

void push(char x)

{

if(t==99)

printf("stack overflow");

else

s[++t]=x;

}

void pop()

{

printf("%c",s[t--]);

}

main()

{

// char str[]="Shru"; COMPILE TIME

char str[100];

printf("Enter any string:\n");

gets(str); // RUN TIME

printf("Reverse string:\n");

int l=strlen(str);

int i;

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

push(str[i]);

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

pop();

}

OUTPUT :

Enter any string:

SKP @ 48

Reverse string:

84 @ PKS

1. PALINDROME :-

CODE :

#include<stdio.h>

#include<string.h>

int t,str[100],flag;

void push(char x)

{

if(t==99)

printf("stack overflow");

else

str[++t]=x;

}

void pop()

{

printf("%c",str[t--]);

}

main()

{

// char str[]="Shru"; COMPILE TIME

char str[100];

printf("Enter any string:\n");

gets(str); // RUN TIME

int l=strlen(str);

printf("l=%d\n",l);

printf("Reverse string:\n");

int i;

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

push(str[i]);

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

pop();

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

{

if(str[i]!=str[l-i-1])

flag=1;

// else

// flag=0;

}

if(flag==1)

printf("\nString is not a Palindrome!");

else

printf("\nString is Palindrome!");

}

OUTPUT :

Enter any string:

Shru @ 48

l=9

Reverse string:

84 @ urhS

String is not a Palindrome!

1. SIZE OF STACK :-

CODE :

#include<stdio.h>

int stack[100],ch,n,top,i,x;

void push(); // nA,nR

void pop();

void display();

void size();

int main()

{

top=-1;

printf("Enter size of stack:\n");

scanf("%d",&n);

printf("1.Push\t2.Pop\t3.Display\t4.Size\t5.Exit\n");

do

{

printf("Enter your choice:\n");

scanf("%d",&ch);

switch(ch)

{

case 1:

push();

break;

case 2:

pop();

break;

case 3:

display();

break;

case 4:

size();

break;

case 5:

printf("Exit point!");

break;

default:

printf("Enter valid choice(1/2/3/4/5)\n");

}

}

while(ch!=5);

return 0;

}

void push()

{

if(top>=n-1)

printf("Stack is OVERFLOW\n");

else

{

printf("Enter a value to be added\n");

scanf("%d",&x);

top++;

stack[top]=x;

}

}

void pop()

{

if(top<=-1)

printf("Stack is UNDERFLOW\n");

else

{

printf("%d is a value to be removed\n",stack[top]);

top--;

}

}

void display()

{

if(top>-1)

{

printf("The elements in stack:\n");

for(i=top;i>=0;i--)

printf("%d\n",stack[i]);

//printf("Enter next choice:\n");

}

else

printf("Stack is EMPTY\n");

}

void size()

{

printf("%d is the size of stack\n",top+1);

}

OUTPUT:

Enter size of stack:

3

1.Push 2.Pop 3.Display 4.Size 5.Exit

Enter your choice:

1

Enter a value to be added

22

Enter your choice:

1

Enter a value to be added

33

Enter your choice:

1

Enter a value to be added

44

Enter your choice:

1

Stack is OVERFLOW

Enter your choice:

2

44 is a value to be removed

Enter your choice:

3

The elements in stack:

33

22

Enter your choice:

4

2 is the size of stack

Enter your choice:

2

33 is a value to be removed

Enter your choice:

2

22 is a value to be removed

Enter your choice:

2

Stack is UNDERFLOW

Enter your choice:

4

0 is the size of stack

Enter your choice:

5

Exit point!

1. TOWER OF HANOI :-

CODE :

#include<stdio.h>

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

int main()

{

int n;

printf("Enter the number of disks\n");

scanf("%d",&n);

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

}

void toh(int n,char Beg,char End,char Aux)

{

if(n>0){

toh(n-1,Beg,Aux,End);

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

toh(n-1,End,Beg,Aux);

}

}

OUTPUT :

Enter the number of disks

3

Move disk 1 from A to C

Move disk 2 from A to B

Move disk 1 from C to B

Move disk 3 from A to C

Move disk 1 from B to A

Move disk 2 from B to C

Move disk 1 from A to C

6.Add a method search (int ob) to the Array Stack. The executor tells how many elements would have to be popped to have ob removed from the stack; it returns -1 if ob is not in the stack.

SEARCH :-

CODE :

#include<stdio.h>

#include<string.h>

int pop();

void push();

int top=-1,n;

char stack[10];

int main()

{

int i,count=0,s;

printf("Enter the size:\n");

scanf("%d",&n);

printf("Enter the elements\n");

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

{

push();

}

printf("Enter the element to search\n");

scanf("%d",&s);

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

{

count++;

if(s == pop())

{

printf("%d times to pop to remove %d\n",count+1,s);

}

else if(count==n)

{

printf("-1");

}

}

}

void push()

{

int x;

if(top>=n-1)

{

printf("Stack is overflow\n");

}

else

{

scanf("%d",&x);

top++;

stack[top]=x;

printf("New element added successfully\n");

}

}

int pop()

{

int c;

c=stack[top];

top--;

return c;

}

OUTPUT :

Enter the size:

5

Enter the elements

1

New element added successfully

2

New element added successfully

3

New element added successfully

4

New element added successfully

5

New element added successfully

Enter the element to search

4

3 times to pop to remove 4

-1

7.You have an empty sequence, and you will be given N queries. Each query is one of these three types:

1 -Push the element x into the stack.

2 -Delete the element present at the top of the stack.

3 -Print the maximum element in the stack.

CODE :

#include<stdio.h>

#include<stdlib.h>

int data[100000],top,max=0;

int main()

{

int t,n,choice;

top=-1;

printf("Enter the size:\n");

scanf("%d",&t);

while(t--)

{

printf("Enter your choice\n1.Push\n2.Pop\n3.Display max no\n");

scanf("%d",&choice);

switch(choice)

{

case 1 : push();

break;

case 2: pop();

break;

case 3: printf("%d\n",max);

break;

}

}

return 0;

}

void push()

{

printf("Enter an element to be pushed:\n");

int item;

scanf("%d",&item);

top++;

data[top]=item;

if(max < data[top])

max = data[top];

}

void pop()

{

int i;

if(max == data[top])

max=0;

top--;

printf("Element is removed\n");

for(i= top;i>=0;i--)

if(max < data[i])

max = data[i];

}

OUTPUT :

Enter your choice

1.Push

2.Pop

3.Display max no

1

Enter an element to be pushed:

44

Enter your choice

1.Push

2.Pop

3.Display max no

1

Enter an element to be pushed:

55

Enter your choice

1.Push

2.Pop

3.Display max no

3

55

Enter your choice

1.Push

2.Pop

3.Display max no

2

Element is removed