

SSN COLLEGE OF ENGINEERING (Autonomous)

DEPARTMENT OF CSE

UCS308 Data Structures Lab

## Assignment 5

### Implementation of Stack

Register Number : 185001131

Name : Sai Charan B

Class : CSE – B

Function.h :

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

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

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

```
#include<math.h>
```

```
typedef struct myStack
```

```
{
```

```
    char data;
```

```
    struct myStack *next;
```

```
}stack;
```

```
int isEmpty(stack *top)
```

```
{
```

```
    return top==NULL;
```

```
}
```

```
char peek(stack *top)
```

```
{
```

```
    return top->data;
```

```
}
```

```
void push(stack **top, char x)
```

```
{
```

```
    stack * temp=(stack*)malloc(sizeof(stack));
```

```
    temp->data=x;
```

```
    if(*top==NULL)
```

```
        temp->next=NULL;
```

```
    else
```

```
        temp->next=*top;

        *top=temp;

    }

char pop(stack **top)

{

    if(isEmpty(*top))

        return '@';

    else

    {

        stack *temp=*top;

        char x=(*top)->data;

        *top=(*top)->next;

        free(temp);

        return x;

    }
```

```
}
```

```
int prec(char ch)
```

```
{
```

```
    if(ch=='^')
```

```
        return 3;
```

```
    else if(ch=='*' || ch=='/')
```

```
        return 2;
```

```
    else if(ch=='+' || ch=='-')
```

```
        return 1;
```

```
    else
```

```
        return -1;
```

```
}
```

```
void evaluate(stack *top2, char s[50])
```

```
{
```

```
    for(int i=0; s[i]; i++)
```

```
{

    if(s[i]>='0'&& s[i]<='9')

        push(&top2,s[i]-'0');

    else

    {

        int x=pop(&top2);

        int y=pop(&top2);

        switch(s[i])

        {

            case '+':

                push(&top2,y+x);

                break;

            case '-':

                push(&top2,y-x);

                break;

            case '*':
```

```

        push(&top2,y*x);

        break;

    case '/':

        push(&top2,y/x);

        break;

    case '^':

        push(&top2,pow(y,x));

        break;

    }

}

}

printf("\nEvaluated expression: %d",pop(&top2));

}

void convert(stack *top,stack * top2,char s[50],char post[50])

{

```

```
int flag=0;

int c=0;

for(int i=0;s[i];i++)

{

    if(s[i]>='0'&& s[i]<='9')

    {

        post[c++]=s[i];

    }

    else if(s[i]=='(')

    {

        push(&top,s[i]);

    }

    else if(s[i]==')')

    {
```

```
while(!isEmpty(top) && peek(top) != '(')

{

    post[c++] = pop(&top);

}

if(isEmpty(top))

{

    flag=1;

    break;

}

else

    pop(&top);

}

else

{
```



```

        while(!isEmpty(top) && prec(s[i])<=prec(peek(top)))

            post[c++]=pop(&top);

        push(&top,s[i]);

    }

}

while(!isEmpty(top))

{

    char ch=pop(&top);

    if(ch=='(')

    {

        flag=1;

        break;

    }

    else

        post[c++]=ch;

}

```

```

        if(flag==1)

            printf("Unbalanced expression\n");

        else

        {

            post[c]='\0';

            printf("Postfix expression: %s",post);

            evaluate(top2,post);

        }

    }
}

```

Main Function:

```

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

#include<math.h>

#include"function.h"

```

```

int main()

{

    stack *top=NULL,*top2=NULL;

    char s[50],post[50];

    printf("Enter expression: ");

    scanf("%s",s);

    convert(top,top2,s,post);

    printf("\n1. Continue \n2. Exit");

    int c;

    scanf("%d",&c);

    if(c==1)

        main();

    else

        return 0;

}

```

## Output:

Enter expression: (2+5)\*(3-6)/(7\*8)

Postfix expression: 25+36-\*78\*/

Evaluated expression: 0

1. Continue

2. Exit1

Enter expression:  $7 - (((3+2) * (6+1)) / (5+6))$

Unbalanced expression

1. Continue

2. Exit1

Enter expression:  $((3+2) * (2+5))$

Unbalanced expression

1. Continue

2. Exit2