CSE 2003

DATA STRUCTURES AND ALGORITHMS



Lab CAT

September 26, 2020

L19+L20 | SJT317

FALL SEMESTER 2020-21

by

SHARADINDU ADHIKARI 19BCE2105

Question 1

Problem:

Write a program to implement a stack using an array and Linked List.

Code, SS & CMD:

Using Linked list:

```
#include<stdio.h>
#include <stdlib.h>
#define TRUE 1
#define FALSE 0
struct node
    int data;
    struct node *next;
typedef struct node node;
node *top;
void initialize()
   top = NULL;
void push(int value)
   node *tmp;
   tmp = malloc(sizeof(node));
   tmp -> data = value;
   tmp -> next = top;
   top = tmp;
}
int pop()
   node *tmp;
   int n;
   tmp = top;
   n = tmp->data;
   top = top->next;
   free(tmp);
   return n;
}
int Top()
    return top->data;
int isempty()
   return top==NULL;
```

```
void display(node *head)
        if(head == NULL)
               printf("NULL\n");
        }
        else
        {
                printf("%d\n", head -> data);
                display(head->next);
        }
}
int main()
        initialize();
       push(10);
       push (20);
       push(30);
       printf("The top is %d\n",Top());
        printf("The top after pop is %d\n",Top());
       display(top);
       return 0;
}
                                                                                                                                                            O
main.c [lab cat] - Code::Blocks 20.03
The Edit (New Search Project Build Debug Fortran wsSmith Tools Tools+ Plugins DoxyBlocks Settings Help

P → ● ● ● ● ● ● □ Debug

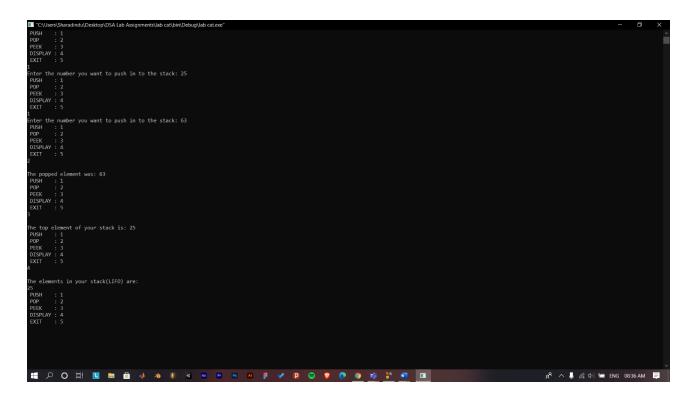
Debug
                                                                              √ | ← → | № № № № | ≈ ▼ | /** *< | ● 🏆 | ♦ | ● | →
 node:
               ~ Q 4
Managemen ×
Project
Workspace
                  struct node
                    int data;
struct node *next;
                  typedef struct node node;
                  void initialize()
                 top = NULL;
                  node *tmp;
tmp = malloc(sizeof(node));
tmp -> data = value;
tmp -> next = top;
top = tmp;
          🕯 🖟 Code:Blocks: X 👊 Search results X 🖟 Coccc 🔣 🛱 Build log X 🔶 Build messages X 🥬 CippCheck/Vera++ X 🥬 CippCheck/Vera++ Ressages X 📝 Cocope X 🟚 Debugger X 🧗 Oxy8locks X 👍 Fortran info X 🔩 Closed files list X
          Boild: Debug in lab cat (compiler: GNU OCC Compiler)

goc.exe -Wall -g -c "C:(Users\Sharadindu\Sesktop\DSA Lab Assignments\lab cat\main.c" -o obj\Sebug\main.o
goc.exe -0 "Dint\Sebug\lab cat:exe" obj\Sebug\main.o
goc.exe -0 "Dint\Sebug\lab cat:exe" obj\Sebug\main.o
Gosput file: si sin\Sebug\lab cat:exe with size $8.74 MB
Process terminated with seaso 0 (a namete(s), o second(s))
O error(s), O warning(s) (0 minute(s), o second(s))
 < >
                                                                       C/C++
                                                                                  Windows (CR+LF) WINDOWS-1252 Line 9, Col 23, Pos 128 Insert
я<sup>R</sup> \land 🎍 🦟 Фі) 🔚 ENG 08:22 АМ 📮
"C:\Users\Sharadindu\Desktop\DSA Lab Assignments\lab cat\bin\Debug\lab cat.exe"
The top is 30
The top after pop is 20
20
10
NULL
Process returned 0 (0x0) execution time: 0.040 s
Press any key to continue.
```

Using Array:

```
#include <stdio.h>
#include <stdlib.h>
#includeocess.h>
int 1[100],t;
void push()
    int p;
   printf("Enter the number you want to push in to the stack: ");
   scanf("%d",&p);
    t=t+1;
   1[t]=p;
}
void pop()
   if(t-1<0)
   printf("\nNo elements to pop.\n");
    else
        printf("\nThe popped element was: ");
        printf("%d", l[t]);
        t=t-1;
   printf("\n");
void peek()
   printf("\nThe top element of your stack is: ");
   printf("%d", 1[t]);
   printf("\n");
void display()
   int i;
   printf ("\nThe elements in your stack(LIFO) are:\n");
   for (i=t;i>=0;i--)
   printf("%d ",l[i]);
   printf("\n");
int main()
{
    int R;
    t=-1;
   printf (" PUSH : 1\n POP : 2\n PEEK : 3\n DISPLAY : 4\n EXIT
5\n");
   scanf("%d",&R);
   while (R!=5)
        if(R==1)
       push();
        else if (R==2)
        pop();
        else if (R==3)
        peek();
        else if (R==4)
        display();
       printf (" PUSH
                       : 1\n POP : 2\n PEEK : 3\n DISPLAY : 4\n EXIT
: 5\n'');
       scanf("%d",&R);
    if (R==5)
```

```
exit(0);
                      return 0;
                      getch();
}
Main.c [lab cat] - Code::Blocks 20.03
                                                                                                                                                                                                                                                                                                                                                                                                                                                    <global>
                                              B
     Project: >
) Workspace
                                                   void push()
                                                           int p; printf("Enter the number you want to push in to the stack: "); scanf("%d", 4p); text=1; [[c]=p;
                                                   void pop()
                                                           if(t-1<0)
printf("\nNo elements to pop.\n");
else</pre>
                                                             f
  printf("\nThe popped element was: ");
  printf("%d",1[t]);
  t=t-1;
                                                           printf("\n");
                                                  void peek()
                                                           printf("\nThe top element of your stack is: ");
printf("%d",1[t]);
printf("\n");
                              -- Build: Debug in lab cat (compiler: GNU GCC Compiler)---
                            Target is up to date.
Nothing to be done (all items are up-to-date)
                                                 ---- Run: Debug in lab cat (compiler: GNU GCC Compiler)-----
   < >
| Read/Write | default | Read/Write | default | Read/Write | default | Read/Write | Read/Write
```



Question 2

Problem:

Write a Program to convert an infix expression to prefix form.

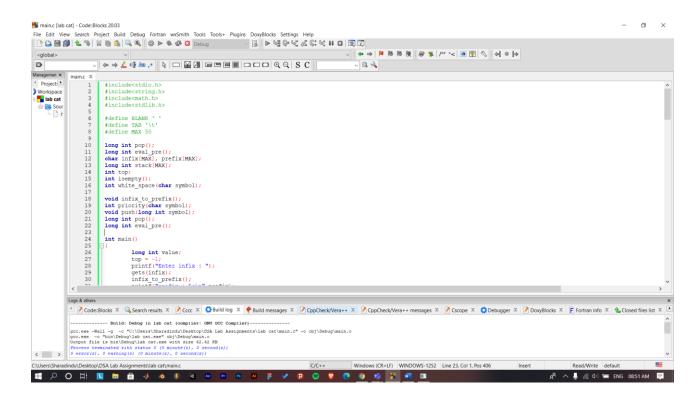
Code, SS & CMD:

```
#include<stdio.h>
#include<string.h>
#include<math.h>
#include<stdlib.h>
#define BLANK ' '
#define TAB '\t'
#define MAX 50
long int pop();
long int eval pre();
char infix[MAX], prefix[MAX];
long int stack[MAX];
int top;
int isempty();
int white_space(char symbol);
void infix to prefix();
int priority(char symbol);
void push(long int symbol);
long int pop();
long int eval pre();
int main()
        long int value;
        top = -1;
        printf("Enter infix : ");
        gets(infix);
        infix_to_prefix();
        printf("prefix : %s\n",prefix);
        value=eval pre();
        printf("Value of expression : %ld\n", value);
        return 0;
}/*End of main()*/
void infix to prefix()
        int i,j,p,n;
        char next ;
        char symbol;
        char temp;
        n=strlen(infix);
        p=0;
        for (i=n-1; i>=0; i--)
                symbol=infix[i];
                if(!white space(symbol))
                 {
```

```
switch(symbol)
                        case ')':
                                push(symbol);
                                break;
                        case '(':
                                while( (next=pop()) != ')')
                                        prefix[p++] = next;
                                break;
                        case '+':
                        case '-':
                        case '*':
                        case '/':
                        case '%':
                        case '^':
                                while( !isempty() && priority(stack[top])>
priority(symbol) )
                                        prefix[p++] = pop();
                                push(symbol);
                                break;
                        default: /*if an operand comes*/
                             prefix[p++] = symbol;
                }
        while(!isempty())
                prefix[p++] = pop();
        prefix[p] = '\0'; /*End prefix with'\0' to make it a string*/
        for (i=0, j=p-1; i < j; i++, j--)
                temp=prefix[i];
                prefix[i]=prefix[j];
                prefix[j]=temp;
}/*End of infix to prefix()*/
/* This function returns the priority of the operator */
int priority(char symbol )
{
        switch(symbol)
        {
        case ')':
              return 0;
        case '+':
        case '-':
               return 1;
        case '*':
        case '/':
        case '%':
               return 2;
        case '^':
                return 3;
        default :
                 return 0;
        }/*End of switch*/
}/*End of priority()*/
void push(long int symbol)
        if(top > MAX)
                printf("Stack overflow\n");
                exit(1);
```

```
}
        else
                top=top+1;
                stack[top] = symbol;
}/*End of push()*/
long int pop()
        if(top == -1)
                printf("Stack underflow \n");
                exit(2);
        return (stack[top--]);
}/*End of pop()*/
int isempty()
        if(top==-1)
                return 1;
        else
                return 0;
}
int white space(char symbol)
        if(symbol==BLANK || symbol==TAB || symbol=='\0')
                return 1;
        else
                return 0;
}/*End of white space()*/
long int eval pre()
        long int a,b,temp,result;
        int i;
        for(i=strlen(prefix)-1;i>=0;i--)
                if(prefix[i] <= '9' && prefix[i] >= '0')
                         push( prefix[i]-48 );
                else
                 {
                         b=pop();
                         a=pop();
                         switch(prefix[i])
                         case '+':
                                 temp=b+a; break;
                         case '-':
                                 temp=b-a;break;
                         case '*':
                                 temp=b*a;break;
                         case '/':
                                 temp=b/a;break;
                         case '%':
                                 temp=b%a;break;
                         case '^':
                                 temp=pow(b,a);
                         }
                         push(temp);
                }
        }
```

```
result=pop();
return result;
}/*End of eval_pre */
```



```
■ "C:\Users\Sharadindu\Desktop\DSA Lab Assignments\lab cat\bin\Debug\lab cat.exe"

Enter infix : A + B + C + D

prefix : +++ABCD

Stack underflow

Process returned 2 (0x2) execution time : 19.609 s

Press any key to continue.
```