

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());
    pop();
    printf("The top after pop is %d\n", Top());
    display(top);
    return 0;
}

```

The screenshot shows the Code::Blocks IDE with the following components:

- Source Code Editor:** Displays the implementation of the linked list operations. It includes headers, struct definitions, and functions for initialization, pushing, popping, and displaying the list.
- Output Window:** Shows the build process using GCC and the execution of the program. The output matches the expected behavior: "The top is 30", "The top after pop is 20", "20", "10", "NULL", and "Process returned 0 (0x0) execution time : 0.040 s".

```

"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>
#include <process.h>
int l[100],t;

void push()
{
    int p;
    printf("Enter the number you want to push in to the stack: ");
    scanf("%d",&p);
    t=t+1;
    l[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",l[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
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
<global> pop0: void
main.c X
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <process.h>
4 int l[100], t;
5
6 void push()
7 {
8     int p;
9     printf("Enter the number you want to push in to the stack: ");
10    scanf("%d", &p);
11    t=t+1;
12    l[t]=p;
13 }
14 void pop()
15 {
16     if(t<=0)
17         printf("\nNo elements to pop.\n");
18     else
19     {
20         printf("\nThe popped element was: ");
21         printf("%d", l[t]);
22         t=t-1;
23     }
24     printf("\n");
25 }
26 void peek()
27 {
28     printf("\nThe top element of your stack is: ");
29     printf("%d", l[t]);
30     printf("\n");
31 }

```

Logs & others

Code::Blocks Search results Cccc Build log Build messages CppCheck/Vera++ CppCheck/Vera++ messages Cscope Debugger DoxyBlocks Fortran info Closed files list

----- 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)-----

Open an existing file C/C++ Windows (CR+LF) WINDOWS-1252 Line 24, Col 18, Pos 426 Insert Read/Write default ENG 08:38 AM

```

"C:\Users\Sharadindu\Desktop\DSA Lab Assignments\lab cat\bin\Debug\lab cat.exe"
PUSH : 1
POP : 2
PEEK : 3
DISPLAY : 4
EXIT : 5
1
Enter the number you want to push in to the stack: 25
PUSH : 1
POP : 2
PEEK : 3
DISPLAY : 4
EXIT : 5
2
The popped element was: 63
PUSH : 1
POP : 2
PEEK : 3
DISPLAY : 4
EXIT : 5
3
The top element of your stack is: 25
PUSH : 1
POP : 2
PEEK : 3
DISPLAY : 4
EXIT : 5
4
The elements in your stack(LIFO) are:
25
PUSH : 1
POP : 2
PEEK : 3
DISPLAY : 4
EXIT : 5

```

Windows taskbar at the bottom shows the time as 08:38 AM and the language as ENG.

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 */

```

The screenshot shows the Code::Blocks 20.03 IDE. The main editor displays the source code for 'main.c'. The code includes standard headers, defines constants, and implements functions for evaluating infix and prefix expressions. The output window at the bottom shows the build process and the execution of the program.

```

main.c [lab cat] - Code::Blocks 20.03
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
<global>
Project: main.c
Workspace: lab cat
Source Files: main.c

1 #include<stdio.h>
2 #include<string.h>
3 #include<math.h>
4 #include<stdlib.h>
5
6 #define BLANK ' '
7 #define TAB '\t'
8 #define MAX 50
9
10 long int pop();
11 long int eval_pre();
12 char infix[MAX], prefix[MAX];
13 long int stack[MAX];
14 int top;
15 int isempty();
16 int white_space(char symbol);
17
18 void infix_to_prefix();
19 int priority(char symbol);
20 void push(long int symbol);
21 long int pop();
22 long int eval_pre();
23
24 int main()
25 {
26     long int value;
27     top = -1;
28     printf("Enter infix : ");
29     gets(infix);
30     infix_to_prefix();
31 }

Build: Build: Debug in lab cat (compiler: GNU GCC Compiler)-----
gcc.exe -Wall -g -o "C:\Users\Sharadindu\Desktop\DSA Lab Assignments\lab cat\main.o" -o obj\Debug\main.o
gcc.exe -o "bin\Debug\lab cat.exe" obj\Debug\main.o
Output file is bin\Debug\lab cat.exe with size 42.42 KB
Process terminated with status 0 (0 minute(s), 0 second(s))
0 error(s), 0 warning(s), 0 minute(s), 0 second(s)

C:\Users\Sharadindu\Desktop\DSA Lab Assignments\lab cat\main.c C/C++ Windows (CR+LF) WINDOWS-1252 Line 23, Col 1, Pos 406 Insert Read/Write default ENG 08:51 AM

```

"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.

```