

22.write a c program to implement single linked list with insertion and deletion at begin,middle and last?

**PROGRAM:**

```
#include <stdio.h>

#include <stdlib.h>

struct Node {
    int data;
    struct Node* next;
};

void insert(struct Node** head, int value) {
    struct Node* newNode = (struct Node*)malloc(sizeof(struct Node));
    newNode->data = value;
    newNode->next = NULL;
    if (*head == NULL) {
        *head = newNode;
    } else {
        struct Node* current = *head;
        while (current->next != NULL) {
            current = current->next;
        }
        current->next = newNode;
    }
}

void deleteElement(struct Node** head, int value) {
    if (*head == NULL) {
        printf("List is empty.\n");
        return;
    }
    struct Node* current = *head;
    struct Node* previous = NULL;
```

```

while (current != NULL && current->data != value) {
    previous = current;
    current = current->next;
}
if (current == NULL) {
    printf("Element not found in the list.\n");
    return;
}
if (previous == NULL) {
    *head = current->next;
} else {
    previous->next = current->next;
}
free(current);
printf("Element deleted successfully.\n");
}

void display(struct Node* head) {
    struct Node* current = head;
    while (current != NULL) {
        printf("%d ", current->data);
        current = current->next;
    }
    printf("\n");
}

int main() {
    struct Node* head = NULL;
    int choice, value;
    while (1) {
        printf("1. Insert an element\n");

```

```
printf("2. Delete an element\n");
printf("3. Display the list\n");
printf("4. Exit\n");
printf("Enter your choice: ");
scanf("%d", &choice);
switch (choice) {
    case 1:
        printf("Enter the value to insert: ");
        scanf("%d", &value);
        insert(&head, value);
        break;
    case 2:
        printf("Enter the value to delete: ");
        scanf("%d", &value);
        deleteElement(&head, value);
        break;
    case 3:
        display(head);
        break;
    case 4:
        exit(0);
    default:
        printf("Invalid choice. Please try again.\n");}}
return 0;
}
```

## OUTPUT:

```
C:\Users\reddy\OneDrive\Documents\SINGLELINKED.cpp - [Executing] - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
(globals)
Project Classes Debug
SINGLELINKED.cpp
55 while (1) {
56     printf("1. Insert an element\n");
57     printf("2. Delete an element\n");
58     printf("3. Display the list\n");
59     printf("4. Exit\n");
60     printf("Enter your choice: ");
61     scanf("%d", &choice);
62     switch (choice) {
63     case 1:
64         printf("Enter the value to insert: ");
65         scanf("%d", &value);
66         insert(&head, value);
67         break;
68     case 2:
69         printf("Enter the value to delete: ");
70         scanf("%d", &value);
71         deleteElement(&head, value);
72         break;
73     case 3:
74         display(head);
75         break;
76     case 4:
77         break;
78     }
79 }
80
81 Compiler Resources Compile Log Debug Find Results Close
Compilation results...
- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\reddy\OneDrive\Documents\SINGLELINKED.exe
- Output Size: 130.5546875 KiB
- Compilation Time: 0.19s
Line: 82 Col: 1 Sel: 0 Lines: 82 Length: 2245 Insert Done parsing in 0.062 seconds
33°C Partly sunny
```

```
C:\Users\reddy\OneDrive\Documents\SINGLELINKED.exe
1. Insert an element
2. Delete an element
3. Display the list
4. Exit
Enter your choice: 1
Enter the value to insert: 1
1. Insert an element
2. Delete an element
3. Display the list
4. Exit
Enter your choice: 1
Enter the value to insert: 2
1. Insert an element
2. Delete an element
3. Display the list
4. Exit
Enter your choice: 1
Enter the value to insert: 3
1. Insert an element
2. Delete an element
3. Display the list
4. Exit
Enter your choice: 2
Enter the value to delete: 3
```

```
C:\Users\reddy\OneDrive\Documents\SINGLELINKED.cpp - [Executing] - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
(globals)
Project Classes Debug
SINGLELINKED.cpp
55 while (1) {
56     printf("1. Insert an element\n");
57     printf("2. Delete an element\n");
58     printf("3. Display the list\n");
59     printf("4. Exit\n");
60     printf("Enter your choice: ");
61     scanf("%d", &choice);
62     switch (choice) {
63     case 1:
64         printf("Enter the value to insert: ");
65         scanf("%d", &value);
66         insert(&head, value);
67         break;
68     case 2:
69         printf("Enter the value to delete: ");
70         scanf("%d", &value);
71         deleteElement(&head, value);
72         break;
73     case 3:
74         display(head);
75         break;
76     case 4:
77         break;
78     }
79 }
80
81 Compiler Resources Compile Log Debug Find Results Close
Compilation results...
- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\reddy\OneDrive\Documents\SINGLELINKED.exe
- Output Size: 130.5546875 KiB
- Compilation Time: 0.19s
Line: 82 Col: 1 Sel: 0 Lines: 82 Length: 2245 Insert Done parsing in 0.062 seconds
33°C Partly sunny
```

```
C:\Users\reddy\OneDrive\Documents\SINGLELINKED.exe
Enter your choice: 1
Enter the value to insert: 3
1. Insert an element
2. Delete an element
3. Display the list
4. Exit
Enter your choice: 1
Enter the value to insert: 3
1. Insert an element
2. Delete an element
3. Display the list
4. Exit
Enter your choice: 2
Enter the value to delete: 3
Element deleted successfully.
1. Insert an element
2. Delete an element
3. Display the list
4. Exit
Enter your choice: 3
1 2 3
1. Insert an element
2. Delete an element
3. Display the list
4. Exit
Enter your choice: 4
Process exited after 29.37 seconds with return value 0
Press any key to continue . . .
```

23.write a c program to implement stack data structure with push,pop and diplay elements?

**PROGRAM:**

```
#include <stdio.h>

#include <stdlib.h>

#define SIZE 5

int top = -1, inp_array[SIZE];

void push();

void pop();

void show();

int main()
{
    int choice;
    while (1)
    {
        printf("\nPerform operations on the stack:");
        printf("\n1.Push the element\n2.Pop the element\n3.Show\n4.End");
        printf("\n\nEnter the choice: ");
        scanf("%d", &choice);
        switch (choice)
        {
            case 1:
                push();
                break;
            case 2:
                pop();
                break;
            case 3:
                show();
                break;
            case 4:
```

```

        exit(0);
    default:
        printf("\nInvalid choice!!");
    }
}
}

void push()
{
    int x;
    if (top == SIZE - 1)
    {
        printf("\nOverflow!!");
    }
    else
    {
        printf("\nEnter the element to be added onto the stack: ");
        scanf("%d", &x);
        top = top + 1;
        inp_array[top] = x;
    }
}

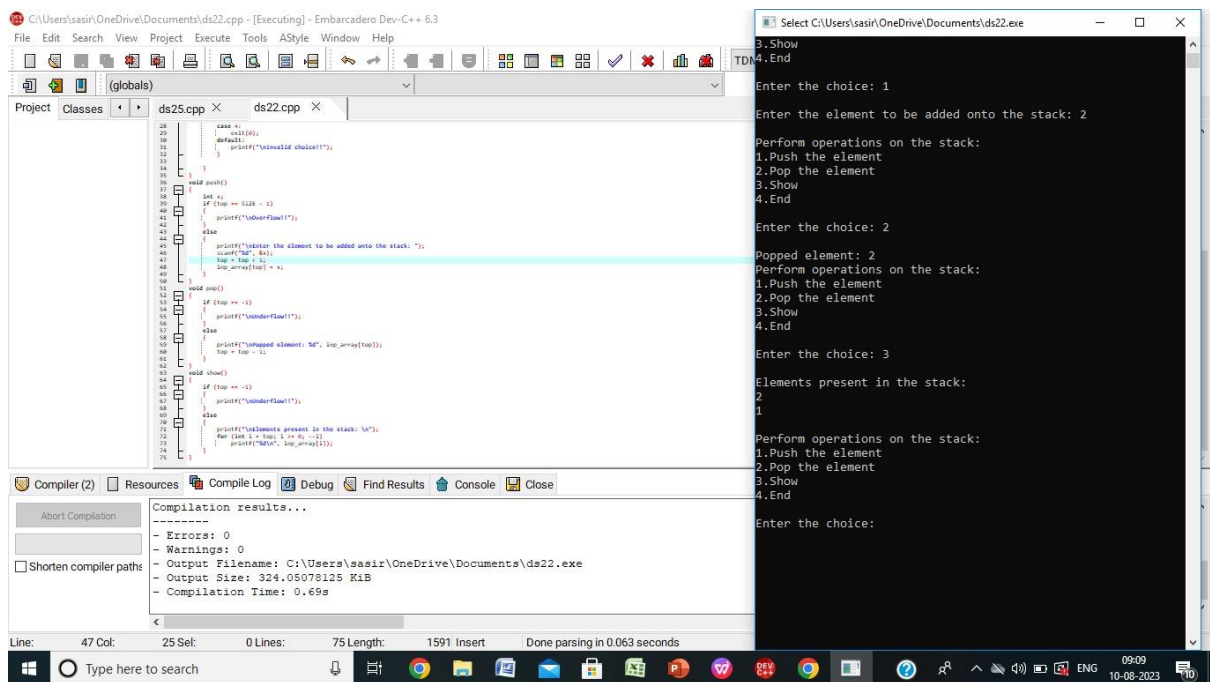
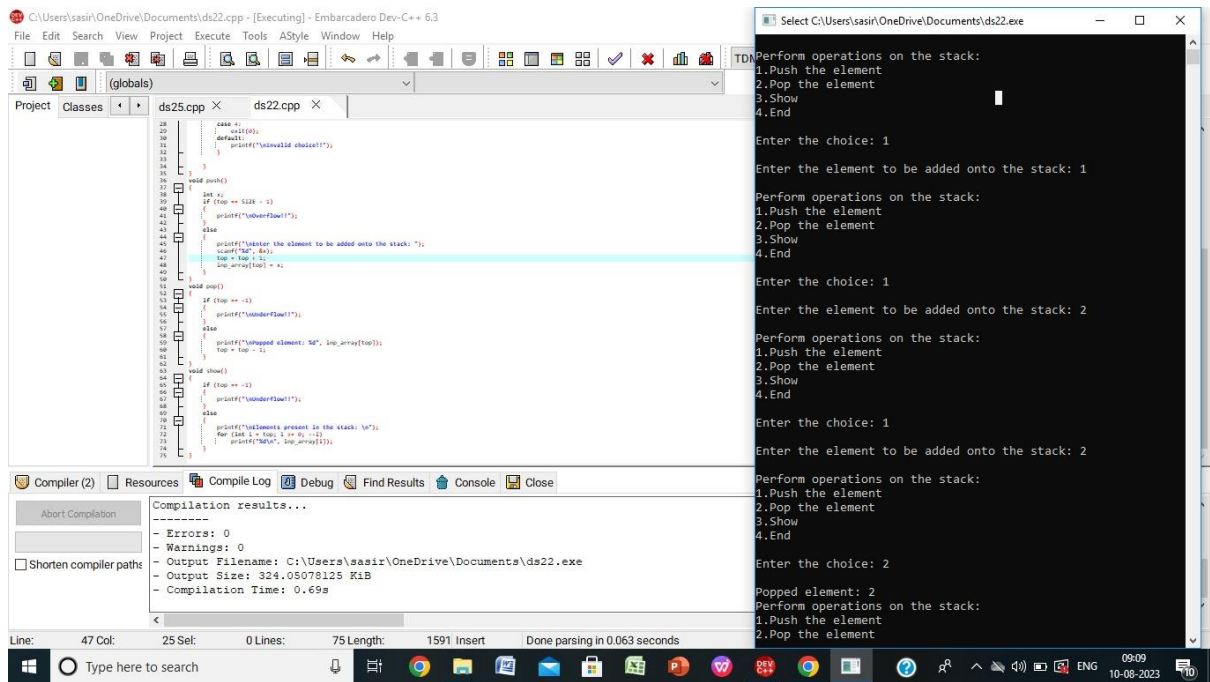
void pop()
{
    if (top == -1)
    {
        printf("\nUnderflow!!");
    }
    else
    {

```

```
        printf("\nPopped element: %d", inp_array[top]);
        top = top - 1;
    }
}

void show()
{
    if (top == -1)
    {
        printf("\nUnderflow!!");
    }
else
    {
        printf("\nElements present in the stack: \n");
        for (int i = top; i >= 0; --i)
            printf("%d\n", inp_array[i]);
    }
}
```

**OUTPUT:**





24.write a c program to implement queue data structure with enqueue,dequeue and display operations?

**PROGRAM:**

```
#include <stdio.h>

# define SIZE 100

void enqueue();
void dequeue();
void show();
int inp_arr[SIZE];
int Rear = - 1;
int Front = - 1;

main()
{
    int ch,exit;
    while (1)
    {
        printf("1.Enqueue Operation\n");
        printf("2.Dequeue Operation\n");
        printf("3.Display the Queue\n");
        printf("4.Exit\n");
        printf("Enter your choice of operations : ");
        scanf("%d", &ch);
        switch (ch)
        {
            case 1:
                enqueue();
                break;
            case 2:
                dequeue();
                break;
```

```

        case 3:
            show();
            break;
        case 4:
                                                    exit;

        default:
            printf("Incorrect choice \n");
    }
}
}

```

```

void enqueue()
{
    int insert_item;
    if (Rear == SIZE - 1)
        printf("Overflow \n");
    else
    {
        if (Front == - 1)

            Front = 0;
        printf("Element to be inserted in the Queue\n : ");
        scanf("%d", &insert_item);
        Rear = Rear + 1;
        inp_arr[Rear] = insert_item;
    }
}

```

```

void dequeue()

```

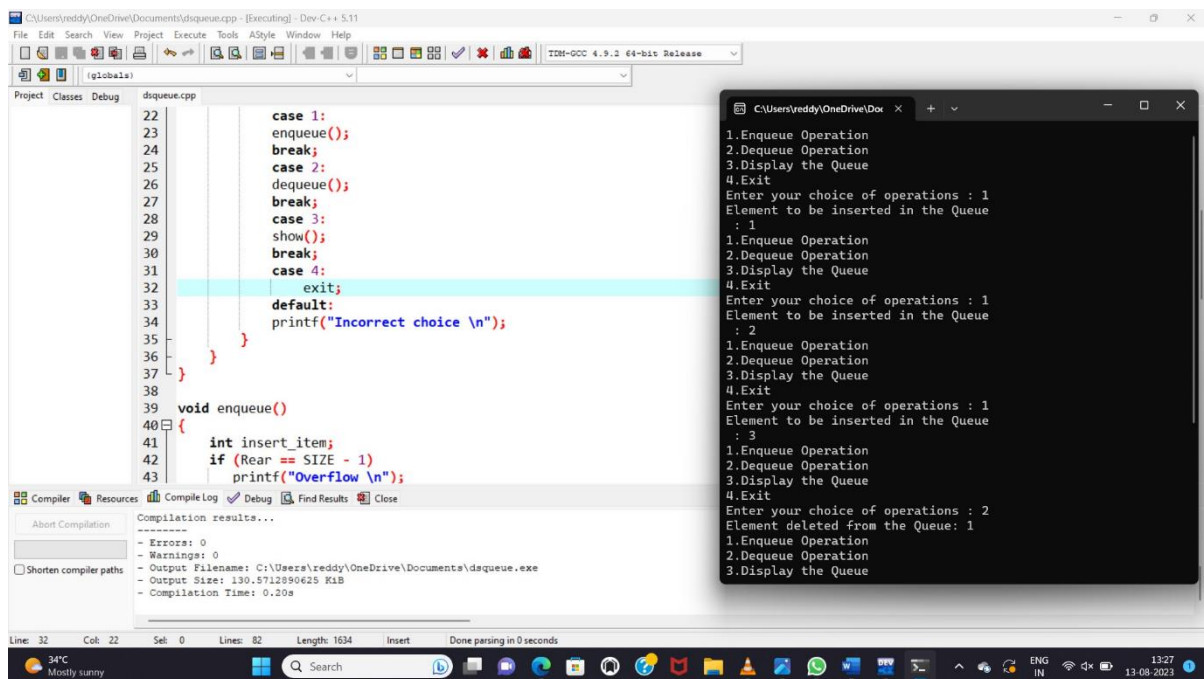
```
{
    if (Front == - 1 || Front > Rear)
    {
        printf("Underflow \n");
        return ;
    }
    else
    {
        printf("Element deleted from the Queue: %d\n", inp_arr[Front]);
        Front = Front + 1;
    }
}
```

void show()

```
{

    if (Front == - 1)
        printf("Empty Queue \n");
    else
    {
        printf("Queue: \n");
        for (int i = Front; i <= Rear; i++)
            printf("%d ", inp_arr[i]);
        printf("\n");
    }
}
```

## OUTPUT:



The screenshot shows a C++ IDE with the file `dsqueue.cpp` open. The code implements a queue with operations: enqueue, dequeue, display, and exit. The `enqueue` function checks for overflow. The terminal window shows the program's output, which includes a menu of operations and user input for choices and elements to be inserted.

```
22     case 1:
23         enqueue();
24         break;
25     case 2:
26         dequeue();
27         break;
28     case 3:
29         show();
30         break;
31     case 4:
32         exit;
33     default:
34         printf("Incorrect choice \n");
35     }
36 }
37
38
39 void enqueue()
40 {
41     int insert_item;
42     if (Rear == SIZE - 1)
43         printf("Overflow \n");
```

Compilation results...

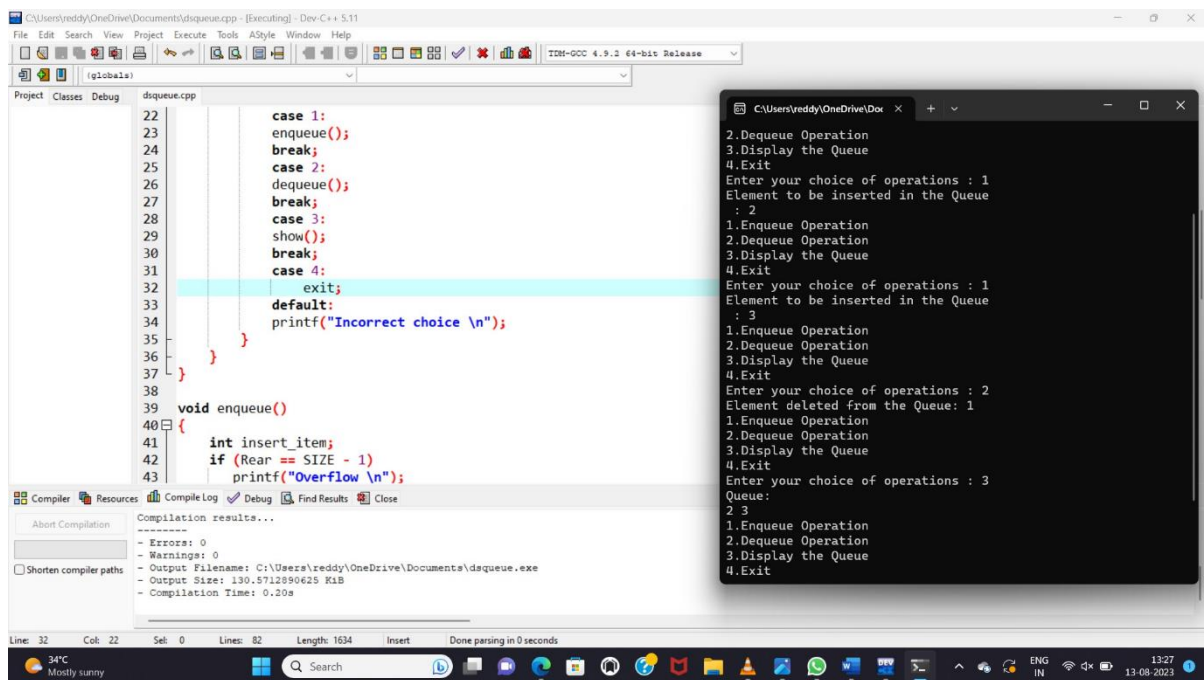
- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\reddy\OneDrive\Documents\dsqueue.exe
- Output Size: 130.5712890625 KiB
- Compilation Time: 0.20s

Line: 32 Col: 22 Sel: 0 Lines: 82 Length: 1634 Insert Done parsing in 0 seconds

34°C Mostly sunny

13:27 13-08-2023

```
1.Enqueue Operation
2.Dequeue Operation
3.Display the Queue
4.Exit
Enter your choice of operations : 1
Element to be inserted in the Queue : 1
1.Enqueue Operation
2.Dequeue Operation
3.Display the Queue
4.Exit
Enter your choice of operations : 1
Element to be inserted in the Queue : 2
1.Enqueue Operation
2.Dequeue Operation
3.Display the Queue
4.Exit
Enter your choice of operations : 1
Element deleted from the Queue: 1
1.Enqueue Operation
2.Dequeue Operation
3.Display the Queue
4.Exit
Enter your choice of operations : 2
Element deleted from the Queue: 1
1.Enqueue Operation
2.Dequeue Operation
3.Display the Queue
4.Exit
Enter your choice of operations : 3
Queue:
2 3
1.Enqueue Operation
2.Dequeue Operation
3.Display the Queue
4.Exit
```



The screenshot shows a C++ IDE with the file `dsqueue.cpp` open. The code implements a queue with operations: enqueue, dequeue, display, and exit. The `enqueue` function checks for overflow. The terminal window shows the program's output, which includes a menu of operations and user input for choices and elements to be inserted.

```
22     case 1:
23         enqueue();
24         break;
25     case 2:
26         dequeue();
27         break;
28     case 3:
29         show();
30         break;
31     case 4:
32         exit;
33     default:
34         printf("Incorrect choice \n");
35     }
36 }
37
38
39 void enqueue()
40 {
41     int insert_item;
42     if (Rear == SIZE - 1)
43         printf("Overflow \n");
```

Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\reddy\OneDrive\Documents\dsqueue.exe
- Output Size: 130.5712890625 KiB
- Compilation Time: 0.20s

Line: 32 Col: 22 Sel: 0 Lines: 82 Length: 1634 Insert Done parsing in 0 seconds

34°C Mostly sunny

13:27 13-08-2023

```
2.Dequeue Operation
3.Display the Queue
4.Exit
Enter your choice of operations : 1
Element to be inserted in the Queue : 2
1.Enqueue Operation
2.Dequeue Operation
3.Display the Queue
4.Exit
Enter your choice of operations : 1
Element deleted from the Queue: 1
1.Enqueue Operation
2.Dequeue Operation
3.Display the Queue
4.Exit
Enter your choice of operations : 3
Queue:
2 3
1.Enqueue Operation
2.Dequeue Operation
3.Display the Queue
4.Exit
```

25.write a c program to convert infix to postfix using stack?

**PROGRAM:**

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

```
#include<ctype.h>
```

```
char stack[100];
```

```
int top = -1;
```

```
void push(char x)
```

```
{  
    stack[++top] = x;  
}
```

```
char pop()
```

```
{  
    if(top == -1)  
        return -1;  
    else  
        return stack[top--];  
}
```

```
int priority(char x)
```

```
{  
    if(x == '(')  
        return 0;  
    if(x == '+' || x == '-')  
        return 1;  
    if(x == '*' || x == '/')  
        return 2;
```

```

        return 0;
    }

int main()
{
    char exp[100];
    char *e, x;
    printf("Enter the expression : ");
    scanf("%s",exp);
    printf("\n");
    e = exp;

    while(*e != '\0')
    {
        if(isalnum(*e))
            printf("%c ",*e);
        else if(*e == '(')
            push(*e);
        else if(*e == ')')
        {
            while((x = pop()) != '(')
                printf("%c ", x);
        }
        else
        {
            while(priority(stack[top]) >= priority(*e))
                printf("%c ",pop());
            push(*e);
        }
    }

```

```

        e++;
    }

    while(top != -1)
    {
        printf("%c ",pop());
    }return 0;
}

```

## OUTPUT:

The screenshot displays a C++ IDE with the source code for an infix-to-postfix converter. The code uses a stack to process the expression 'a b c / d e - \* +'. The output window shows the postfix result 'a b c / d e - \* +' and confirms the process exited successfully after 28.32 seconds.

```

C:\Users\reddy\OneDrive\Documents\infixtopostfix.cpp - [Executing] - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
[Icons] (globals)
Project Classes Debug [?] dsqueue.cpp infixtopostfix.cpp
44         else if(*e == '(')
45             push(*e);
46         else if(*e == ')')
47         {
48             while((x = pop()) != '(')
49                 printf("%c ", x);
50         }
51         else
52         {
53             while(priority(stack[top]) >= priority(*e))
54                 printf("%c ", pop());
55             push(*e);
56         }
57         e++;
58     }
59
60     while(top != -1)
61     {
62         printf("%c ", pop());
63     }return 0;
64
Compiler Resources Compile Log Debug Find Results Close
Abort Compilation
Shorten compiler paths
Compilation results...
- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\reddy\OneDrive\Documents\infixtopostfix.exe
- Output Size: 130.0458994975 KiB
- Compilation Time: 0.22s
Line: 64 Col: 2 Set: 0 Lines: 64 Length: 1077 Insert Done parsing in 0.016 seconds
34°C Mostly sunny
Search
13:43 13-08-2023

```

26.write a c program evaluate given expression using stack?

**PROGRAM:**

```
#include <stdio.h>

#include <ctype.h>

#define MAXSTACK 100

#define POSTFIXSIZE 100

int stack[MAXSTACK];

int top = -1;

void push(int item)
{
    if (top >= MAXSTACK - 1) {
        printf("stack over flow");
        return;
    }
    else {
        top = top + 1;
        stack[top] = item;
    }
}

int pop()
{
    int item;
    if (top < 0) {
        printf("stack under flow");
    }
    else {
        item = stack[top];
        top = top - 1;
        return item;
    }
}
```



```

    }
}

void EvalPostfix(char postfix[])
{
    int i;

    char ch;

    int val;

    int A, B;

    for (i = 0; postfix[i] != '\0'; i++) {
        ch = postfix[i];
        if (isdigit(ch))
            {

                push(ch - '0');
            }
        else if (ch == '+' || ch == '-' || ch == '*' || ch == '/') {
            A = pop();
            B = pop();
            switch (ch)
            {
                case '*':
                    val = B * A;
                    break;
                case '/':
                    val = B / A;
                    break;
                case '+':
                    val = B + A;
                    break;
                case '-':

```

```

        val = B - A;

        break;
    }
    push(val);
}
}

printf(" \n Result of expression evaluation : %d \n", pop());
}

int main()
{
    int i;
    char postfix[POSTFIXSIZE];

    printf("ASSUMPTION: There are only four operators(*, /, +, -) in an expression and
operand is single digit only.\n");

    printf(" \nEnter postfix expression,\npress right parenthesis ')' for end expression : ");

    for (i = 0; i <= POSTFIXSIZE - 1; i++) {
        scanf("%c", &postfix[i]);

        if (postfix[i] == ')')
        {
            break;
        }
    }

    EvalPostfix(postfix);

    return 0;
}

```

## OUTPUT:

```
1 #include <stdio.h>
2 #include <ctype.h>
3 #define MAXSTACK 100
4 #define POSTFIXEXP 100
5 int stack[MAXSTACK];
6 int top = -1;
7 void push(int item)
8 {
9     if (top == MAXSTACK - 1) {
10         printf("stack over flow");
11         return;
12     }
13     else {
14         top = top + 1;
15         stack[top] = item;
16     }
17 }
18 int pop()
19 {
20     int item;
21     if (top < 0) {
22         printf("stack under flow");
23     }
24     else {
25         item = stack[top];
26         top = top - 1;
27         return item;
28     }
29 }
30 void EvalPostfix(char postfix[])
31 {
32     int i;
33     char ch;
34     int val;
35     int A, B;
36     for (i = 0; postfix[i] != '\0'; i++) {
37         ch = postfix[i];
38         if (isdigit(ch)) {
39             val = val * 10 + (ch - '0');
40         }
41         else if (ch == '+' || ch == '-' || ch == '*' || ch == '/') {
42             B = pop();
43             A = pop();
44             if (ch == '+') val = A + B;
45             else if (ch == '-') val = A - B;
46             else if (ch == '*') val = A * B;
47             else if (ch == '/') val = A / B;
48             push(val);
49         }
50     }
51     val = pop();
52     printf("Result of expression evaluation : %d", val);
53 }
```

```
expression and operand is single digit only.
Enter postfix expression,
press right parenthesis ')' for end expression : 524*+1-
)
Result of expression evaluation : 12
-----
Process exited after 35.16 seconds with return value 0
Press any key to continue . . .
```

Compiler (2) | Resources | Compile Log | Debug | Find Results | Console | Close

Line	Col	File	Message
29	1	C:\Users\sasir\OneDrive\Documents\ds26.c...	In function 'int pop()':
		C:\Users\sasir\OneDrive\Documents\ds26.cpp	[Warning] control reaches end of non-void function [-Wreturn-type]

Line: 61 Col: 10 Sel: 0 Lines: 81 Length: 1815 Insert Done parsing in 0.047 seconds

Type here to search

10:53 10-08-2023