

Q1. Stack applications

- i) Infix to Postfix Conversion
- ii) Evaluation of Postfix expression
- iii) Balancing Symbols

i) Infix to Postfix Conversion

Program Code:

```
// C program to convert infix expression to postfix
#include <stdio.h>
#include <string.h>
#include <stdlib.h>

// Stack type
typedef struct
{
    int top;
    int stack[200];
}Stack;

// Stack Operations
int new_stack( Stack *stack )
{
    stack->top = -1;
}
int empty(Stack* stack)
{
    return stack->top == -1 ;
}
char peek(Stack* stack)
{
    return stack->stack[stack->top];
}
char pop(Stack* stack)
{
    if (!empty(stack))
        return stack->stack[stack->top--] ;
    printf("Stack underflow!");
}
void push(Stack* stack, char op)
{
    stack->stack[++stack->top] = op;
}

int is_operand(char ch)
{
    return '0'<=ch && ch<='9';
}

int precedence(char ch)
```

```

{
    switch (ch)
    {
        case '+':
        case '-':
            return 1;

        case '*':
        case '/':
            return 2;

        case '^':
            return 3;
    }
    return -1;
}

int main()
{
    char exp[200];
    int i, k;
    Stack stack;
    new_stack(&stack);
    printf("Enter the expression : ");
    fgets(exp, 200, stdin);
    for (i = 0, k = -1; exp[i]; ++i)
    {
        if (is_operand(exp[i]))
            exp[++k] = exp[i];
        else if (exp[i] == '(')
            push(&stack, exp[i]);

        else if (exp[i] == ')')
        {
            while (!empty(&stack) && peek(&stack) != '(')
                exp[++k] = pop(&stack);
            if (!empty(&stack) && peek(&stack) != '(')
                return -1;
            else
                pop(&stack);
        }
        else
        {
            while (!empty(&stack) &&
                   precedence(exp[i]) <= precedence(peek(&stack)))
                exp[++k] = pop(&stack);
            push(&stack, exp[i]);
        }
    }

    while (!empty(&stack))

```

```

exp[++k] = pop(&stack);

exp[++k] = '\0';
printf( "%s", exp );

return 0;
}

```

Program Screenshots:

```

Activities  Visual Studio Code
File Edit Selection View Go Run Terminal Help
queue.c patient.c deque.c M VL2022230103816_AST01.pdf infix_postfix.c - lab_assessment1 - Visual Studio Code
Sun Aug 21 22:34
1 // C program to convert infix expression to postfix
2 #include <stdio.h>
3 #include <string.h>
4 #include <stdlib.h>
5
6 // Stack type
7 You, 19 minutes ago | 1 author (You)
8 typedef struct
9 {
10     int top;
11     int stack[200];
12 }Stack;
13
14 // Stack Operations
15 int new_stack( Stack *stack )
16 {
17     stack->top = -1;
18 }
19 int empty(Stack* stack) You, 3 hours ago + feat: added stuff
20 {
21     return stack->top == -1 ;
22 }
23 char peek(Stack* stack)
24 {
25     return stack->stack[stack->top];
26 }
27 char pop(Stack* stack)
28 {
29     if (!empty(stack))
30         return stack->stack[stack->top--] ;
31     printf("Stack underflow!");
32 }
33 void push(Stack* stack, char op)
34 {
35     stack->stack[++stack->top] = op;
36 }
37 int is_operand(char ch)
38 {
39     return '0'<=ch && ch<='9';
40 }
41
42 int precedence(char ch)
43 {
44     switch (ch)
45     {
46     case '+':
47     case '-':
48         return 1;
49     case '*':
50     case '/':
51         return 2;
52     case '^':
53         return 3;
54     }
55     return -1;
56 }
57
58 }
59
60 int main()
61 {
62     char exp[200];

```

You, 3 hours ago Ln 18, Col 24 Tab Size: 4 UTF-8 LF C Linux

Activities < Visual Studio Code Sun Aug 21 22:35 infix_postfix.c - lab_assessment1 - Visual Studio Code

```

queue.c patient.c deque.c VL2022230103816_AST01.pdf infix_postfix.c M infix_rpn.c
  C infix_postfix.c > emptyStack()
  int main()
  {
    char exp[200];
    int i, k;
    Stack stack;
    new_stack(&stack);
    printf("Enter the expression : ");
    fgets(exp, 200, stdin);
    for (i = 0, k = -1; exp[i]; ++i)
    {
      if (is_operand(exp[i]))
        exp[++k] = exp[i];
      else if (exp[i] == '(')
        push(&stack, exp[i]);
      else if (exp[i] == ')')
      {
        while (!empty(&stack) && peek(&stack) != '(')
          exp[++k] = pop(&stack);
        if (!empty(&stack) && peek(&stack) != ')')
          return -1;
        else
          pop(&stack);
      }
      else
      {
        while (!empty(&stack) && precedence(exp[i]) <= precedence(peek(&stack)))
          exp[++k] = pop(&stack);
        push(&stack, exp[i]);
      }
    }
    return -1;
  }
  else
    pop(&stack);
}
else
{
  while (!empty(&stack) && precedence(exp[i]) <= precedence(peek(&stack)))
    exp[++k] = pop(&stack);
  push(&stack, exp[i]);
}
while (!empty(&stack))
  exp[++k] = pop(&stack );
exp[++k] = '\0';
printf("Postfix form: %s\n", exp );
  You, 1 second ago • Uncommitted changes
  return 0;
}

```

You, 3 hours ago Ln 18, Col 24 Tab Size: 4 UTF-8 LF C ↻ Linux ↻

Activities < Visual Studio Code Sun Aug 21 22:41 infix_postfix.c - lab_assessment1 - Visual Studio Code

```

queue.c patient.c deque.c VL2022230103816_AST01.pdf infix_postfix.c M infix_rpn.c
  C infix_postfix.c > main()
  return 0;
}

```

You, 1 second ago Ln 98, Col 28 Tab Size: 4 UTF-8 LF C ↻ Linux ↻

Output:

```
[kittycat@fedora lab_assessment1]$ gcc infix_postfix.c && ./a.out
Enter the expression : 2+3/4*5^7
Postfix form: 234/57^*+
```

```
[kittycat@fedora lab_assessment1]$ gcc infix_postfix.c && ./a.out
Enter the expression : (2+3)*(4+6)
Postfix form: 23+46+*
```

```
[kittycat@fedora lab_assessment1]$
```

Output Screenshot:

The screenshot shows the Visual Studio Code interface with the terminal tab selected. The terminal window displays the execution of a C program to evaluate postfix expressions. The user enters expressions like "2+3/4*5^7" and "(2+3)*(4+6)", and the program outputs the postfix form of these expressions.

```
[kittycat@fedora lab_assessment1]$ gcc infix_postfix.c && ./a.out
Enter the expression : 2+3/4*5^7
Postfix form: 23+46*5^4

[kittycat@fedora lab_assessment1]$ (2+3)*(4+6)
Postfix form: 23+46*
```

Below the terminal, the status bar shows the file is main.c, has 0:1 lines, 0△0 errors, and 5 hrs 46 mins since the last save. The bottom right corner shows the user is You, 1 second ago, in Linux, with tab size 4, LF, and C.

ii) Evaluation of Postfix expression:

Program Code:

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

typedef struct
{
    int stack[200];
    int top;
    int size;
} Stack;

int new_stack(Stack *stack, int size)
{
    stack->size = size;
    stack->top = -1;
}

int disp(Stack *stack)
{
    for (int i = stack->top; i >= 0; i--)
    {
        printf("%d ", stack->stack[i]);
    }
    printf("\n");
}

int is_operand(char ch){
    return '0'<=ch && ch<='9';
}

int push(Stack *stack, char val)
{
    stack->top++;
    stack->stack[stack->top] = val;
}
```

```

}

int pop(Stack *stack)
{
    if (stack->top < 0)
    {
        printf("Stack Underflow!\n");
        return 0;
    }
    int idx = stack->top--;
    return stack->stack[idx];
}

int main()
{
    char rpn[200];
    Stack stack;
    new_stack(&stack,50);
    printf("Enter the expression : ");
    for (int i = 0; rpn[i]; i++)
    {
        if (is_operand(rpn[i]))
        {
            push(&stack,rpn[i]-'0');
        }
        else{
            int a;
            switch (rpn[i])
            {
                case '+':
                    a=pop(&stack)+pop(&stack);
                    push(&stack,a);
                    break;
                case '-':
                    a=pop(&stack)-pop(&stack);
                    push(&stack,a);
                    break;
                case '*':
                    a=pop(&stack)*pop(&stack);
                    push(&stack,a);
                    break;
                case '/':
                    a=pop(&stack)/pop(&stack);
                    push(&stack,a);
                    break;
                default:
                    break;
            }
        }
    }
    printf("Solution is: %d\n",pop(&stack));
}

```

Program Screenshots:

Sun Aug 21 22:45 infix_rpn.c - lab_assessment1 - Visual Studio Code

```
queue.c patient.c deque.c VL2022230103816_AST01.pdf infix_postfix.c infix_rpn.c

1 #include <stdio.h>
2 #include <stdlib.h>
3 You, last week | 1 author (You)
4 typedef struct
5 {
6     int stack[200];
7     int top;
8     int size;
9 } Stack;
10 int new_stack(Stack *stack, int size)
11 {
12     stack->size = size;
13     stack->top = -1;
14 }
15 int disp(Stack *stack)    You, last week + feat: added rpn
16 {
17     for (int i = stack->top; i >= 0; i--)
18     {
19         printf("%d ", stack->stack[i]);
20     }
21     printf("\n");
22 }
23
24 int is_operand(char ch){
25     return '0'<=ch && ch<='9';
26 }
27
28 int push(Stack *stack, char val)
29 {
30     stack->top++;
31     stack->stack[stack->top] = val;
32 }
33 int pop(Stack *stack)
34 {
35     if (stack->top < 0)
36     {
37         printf("Stack Underflow!\n");
38         return 0;
39     }
40     int idx = stack->top--;
41     return stack->stack[idx];
42 }
43
44 int main()
45 {
46     char rpn[200];
47     Stack stack;
48     new_stack(&stack,50);
49     printf("Enter the expression : ");    You, 1 second ago + Uncommitted changes
50     fgets(rpn, 200, stdin);
51     for (int i = 0; rpn[i]; i++)
52     {
53         if (is_operand(rpn[i]))
54         {
55             push(&stack,rpn[i]-'0');
56         }
57     }
58 }
```

You, last week Ln 15, Col 23 Spaces:4 UTF-8 LF C Linux

Sun Aug 21 22:50 infix_rpn.c - lab_assessment1 - Visual Studio Code

```
queue.c patient.c deque.c VL2022230103816_AST01.pdf infix_postfix.c infix_rpn.c

24 int is_operand(char ch){
25     return '0'<=ch && ch<='9';
26 }
27
28 int push(Stack *stack, char val)
29 {
30     stack->top++;
31     stack->stack[stack->top] = val;
32 }
33 int pop(Stack *stack)
34 {
35     if (stack->top < 0)
36     {
37         printf("Stack Underflow!\n");
38         return 0;
39     }
40     int idx = stack->top--;
41     return stack->stack[idx];
42 }
43
44 int main()
45 {
46     char rpn[200];
47     Stack stack;
48     new_stack(&stack,50);
49     printf("Enter the expression : ");    You, 1 second ago + Uncommitted changes
50     fgets(rpn, 200, stdin);
51     for (int i = 0; rpn[i]; i++)
52     {
53         if (is_operand(rpn[i]))
54         {
55             push(&stack,rpn[i]-'0');
56         }
57     }
58 }
```

You, 1 second ago Ln 48, Col 39 Spaces:4 UTF-8 LF C Linux

Sun Aug 21 22:50 infix_rpn.c - lab_assessment1 - Visual Studio Code

```
queue.c patient.c deque.c VL2022230103816_AST01.pdf infix_postfix.c infix_rpn.c

43
44 int main()
45 {
46     char rpn[200];
47     Stack stack;
48     new_stack(&stack,50);
49     printf("Enter the expression : ");    You, 1 second ago + Uncommitted changes
50     fgets(rpn, 200, stdin);
51     for (int i = 0; rpn[i]; i++)
52     {
53         if (is_operand(rpn[i]))
54         {
55             push(&stack,rpn[i]-'0');
56         }
57         else{
58             int a;
59             switch (rpn[i])
60             {
61                 case '+':
62                     a=pop(&stack)+pop(&stack);
63                     push(&stack,a);
64                     break;
65                 case '-':
66                     a=pop(&stack)-pop(&stack);
67                     push(&stack,a);
68                     break;
69                 case '*':
70                     a=pop(&stack)*pop(&stack);
71                     push(&stack,a);
72                     break;
73                 case '/':
74                     a=pop(&stack)/pop(&stack);
75                     push(&stack,a);
76             }
77         }
78     }
79 }
```

You, 1 second ago Ln 48, Col 39 Spaces:4 UTF-8 LF C Linux

Activities Visual Studio Code Sun Aug 21 22:51 infix_rpn.c - lab_assessment1 - Visual Studio Code

```
queue.c patient.c deque.c VL2022230103816_AST07.pdf infix_postfix.c infix_rpn.c
```

```
infix_rpn.c > main()
58     switch (rpn[i])
59     {
60         case '+':
61             a=pop(&stack)+pop(&stack);
62             push(&stack,a);
63             break;
64         case '-':
65             a=pop(&stack)-pop(&stack);
66             push(&stack,a);
67             break;
68         case '*':
69             a=pop(&stack)*pop(&stack);
70             push(&stack,a);
71             break;
72         case '/':
73             a=pop(&stack)/pop(&stack);
74             push(&stack,a);
75             break;
76         default:
77             break;
78     }
79 }
80 printf("Solution is: %d\n",pop(&stack));
81 }
```

You, 1 second ago Ln 48, Col 39 Spaces: 4 UTF-8 C Linux

Output:

```
[kittycat@fedora lab_assessment1]$ gcc infix_rpn.c && ./a.out
```

```
Enter the expression : 234/57^*+
```

```
Solution is: 36
```

```
[kittycat@fedora lab_assessment1]$ gcc infix_rpn.c && ./a.out
```

```
Enter the expression : 23+46+*
```

```
Solution is: 50
```

```
[kittycat@fedora lab_assessment1]$
```

Output Screenshot:

Activities Visual Studio Code Sun Aug 21 22:56 infix_rpn.c - lab_assessment1 - Visual Studio Code

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER GITLENS bash + ×
```

```
[kittycat@fedora lab_assessment1]$ gcc infix_rpn.c && ./a.out
Solution is: 36
[kittycat@fedora lab_assessment1]$ gcc infix_rpn.c && ./a.out
Enter the expression : 23+46+*
Solution is: 50
[kittycat@fedora lab_assessment1]$
```

You, 3 hours ago Ln 53, Col 10 Spaces: 4 UTF-8 C Linux

iii) Balancing Symbols

Program Code:

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

typedef struct
{
    char stack[200];
    int top;
    int size;
} Stack;

int new_stack(Stack *stack, int size)
{
    stack->size = size;
    stack->top = -1;
}

int disp(Stack *stack)
{
    for (int i = stack->top; i >= 0; i--)
    {
        printf("%c ", stack->stack[i]);
    }
    printf("\n");
}

int push(Stack *stack, char val)
{
    stack->top++;
    stack->stack[stack->top] = val;
}

int pop(Stack *stack)
{
    if (stack->top < 0)
    {
        printf("Stack Underflow!");
        exit(1);
    }
    stack->top--;
}

int main()
{
    Stack stack;
    new_stack(&stack, 20);
    char brackets[200];
    printf("Enter brackets: ");
    fgets(brackets, 200, stdin);
    for (int i = 0; brackets[i] != '\0'; i++)
    {
        if (stack.top > -1)
```

```

    {
        if (stack.stack[stack.top] == '[' && brackets[i] == ']')
            pop(&stack);
        else if (stack.stack[stack.top] == '{' && brackets[i] == '}')
            pop(&stack);
        else if (stack.stack[stack.top] == '(' && brackets[i] == ')')
            pop(&stack);
        continue;
    }

    if (brackets[i] == '(' || brackets[i] == '{' || brackets[i] == '[')
    {
        push(&stack, brackets[i]);
        continue;
    }

    if (stack.top == -1)
    {

        if (brackets[i] == ')' || brackets[i] == '}' || brackets[i] == ']')
        {
            printf("Un Balanced\n");
            return 0;
        }
    }

    if (stack.top == -1)
    {
        printf("Balanced\n");
    }
    else
    {
        printf("Un Balanced\n");
    }
}

```

Program Screenshots:

The image consists of three vertically stacked screenshots of the Visual Studio Code interface, showing the development of a C program named `brackets.c`.

Screenshot 1 (Top): Shows the initial state of the code. The `disp(Stack *)` function is defined, and the `main()` function calls it. The code prints the stack contents when called.

```
brackets.c > disp(Stack *)  
You, 3 days ago | 1 author (You)  
1 #include <stdio.h>  
2 #include <stdlib.h>  
You, last week | 1 author (You)  
3 typedef struct  
4 {  
5     char stack[200];  
6     int top;  
7     int size;  
8 } Stack;  
9  
10 int new_stack(Stack *stack, int size)  
11 {  
12     stack->size = size;  
13     stack->top = -1;  
14 }  
15 int disp(Stack *stack)  
16 {  
17     You, last week • feat: new dsa programs  
18     for (int i = stack->top; i >= 0; i--)  
19     {  
20         printf("%c ", stack->stack[i]);  
21     }  
22     printf("\n");  
23 }  
24 int push(Stack *stack, char val)  
25 {  
26     stack->top++;  
27     stack->stack[stack->top] = val;  
28 }  
29 int pop(Stack *stack)  
30 {  
31 }
```

Screenshot 2 (Middle): The `pop()` function is implemented. It checks if the stack is underflowed (top < 0) and exits if so. Otherwise, it decrements the top pointer.

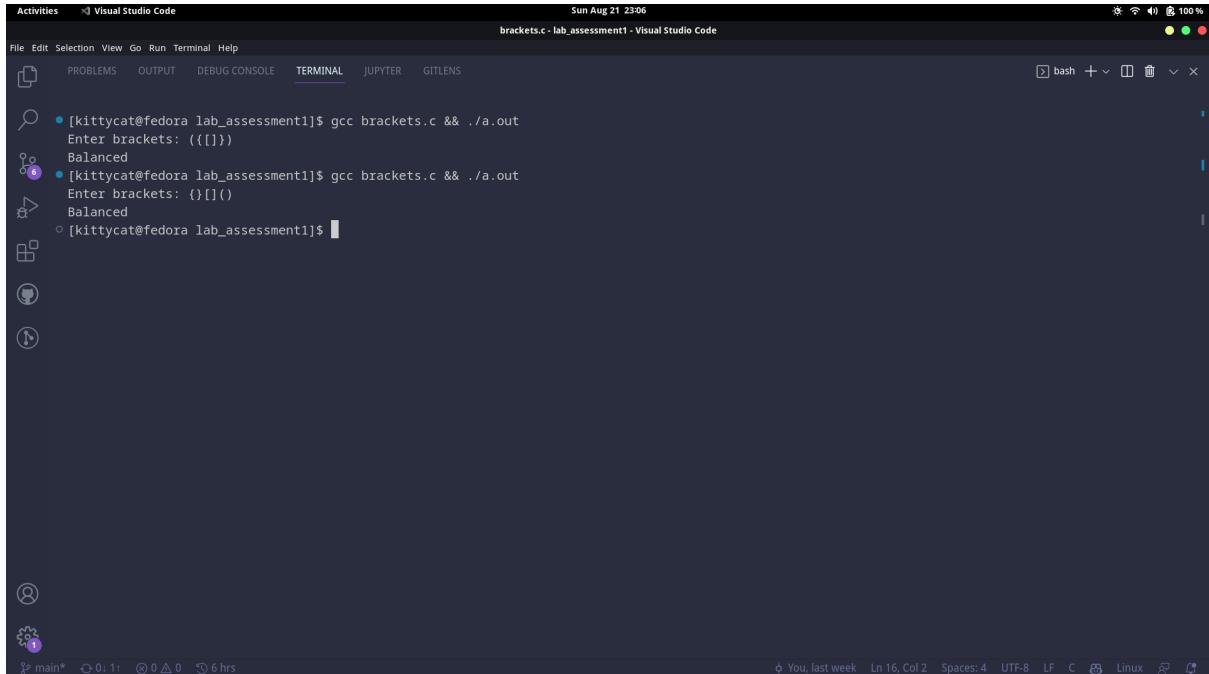
```
brackets.c > disp(Stack *)  
You, last week | 1 author (You)  
29 int pop(Stack *stack)  
30 {  
31     if (stack->top < 0)  
32     {  
33         printf("Stack Underflow!");  
34         exit(1);  
35     }  
36     stack->top--;  
37 }  
38  
39 int main()  
40 {  
41     Stack stack;  
42     new_stack(&stack, 20);  
43     char brackets[200];  
44     printf("Enter brackets: ");  
45     fgets(brackets, 200, stdin);  
46     for (int i = 0; brackets[i] != '\0'; i++)  
47     {  
48         if (stack.top > -1)  
49         {  
50             if (stack.stack[stack.top] == '[' && brackets[i] == ']')  
51                 pop(&stack);  
52             else if (stack.stack[stack.top] == '(' && brackets[i] == ')')  
53                 pop(&stack);  
54             else if (stack.stack[stack.top] == '{' && brackets[i] == '}')  
55                 pop(&stack);  
56             continue;  
57         }  
58  
59         if (brackets[i] == '(' || brackets[i] == '{' || brackets[i] == '[')  
60         {  
61             push(&stack, brackets[i]);  
62             continue;  
63         }  
64         if (stack.top == -1)  
65         {  
66             if (brackets[i] == ')' || brackets[i] == '}' || brackets[i] == ']')  
67                 printf("Un Balanced\n");  
68             return 0;  
69         }  
70     }  
71 }  
72  
73 if (stack.top == -1)  
74 {  
75     printf("Balanced\n");  
76 }  
77 else  
78 {  
79     printf("Un Balanced\n");  
80 }  
81 }  
82 }
```

Screenshot 3 (Bottom): The final state of the code. The `main()` loop now handles all three types of brackets: {}, [], and () correctly, printing "Un Balanced" for unbalanced inputs and "Balanced" for balanced ones.

```
brackets.c > disp(Stack *)  
You, last week | 1 author (You)  
54     else if (stack.stack[stack.top] == '(' && brackets[i] == ')')  
55         pop(&stack);  
56     continue;  
57 }  
58  
59     if (brackets[i] == '(' || brackets[i] == '{' || brackets[i] == '[')  
60     {  
61         push(&stack, brackets[i]);  
62         continue;  
63     }  
64     if (stack.top == -1)  
65     {  
66         if (brackets[i] == ')' || brackets[i] == '}' || brackets[i] == ']')  
67             printf("Un Balanced\n");  
68         return 0;  
69     }  
70 }  
71  
72 if (stack.top == -1)  
73 {  
74     printf("Balanced\n");  
75 }  
76 else  
77 {  
78     printf("Un Balanced\n");  
79 }  
80 }  
81 }  
82 }
```

Output:

```
[kittycat@fedora lab_assessment1]$ gcc brackets.c && ./a.out
Enter brackets: ({[]})
Balanced
[kittycat@fedora lab_assessment1]$ gcc brackets.c && ./a.out
Enter brackets: {}[]()
Balanced
[kittycat@fedora lab_assessment1]$
```

Output Screenshots:

The screenshot shows the Visual Studio Code interface with the terminal tab selected. The terminal window displays the same output as the previous text block, showing the program's response to different bracket inputs. The status bar at the bottom indicates the file is named 'brackets.c - lab_assessment1' and shows basic statistics like lines and characters.

```
[kittycat@fedora lab_assessment1]$ gcc brackets.c && ./a.out
Enter brackets: ({[]})
Balanced
[kittycat@fedora lab_assessment1]$ gcc brackets.c && ./a.out
Enter brackets: {}[]()
Balanced
[kittycat@fedora lab_assessment1]$
```

2. Write a menu driven program to perform static implementation of a queue data structure with all possible functions.**Program Code:**

```
#include "stdio.h"

typedef struct
{
    int queue[200];
    int front;
    int rear;
} Queue;

int new (Queue *queue)
{
    queue->front = -1;
    queue->rear = -1;
}

int enqueue(Queue *queue, int val)
{
    if (queue->front > queue->rear)
    {
        printf("Queue Overflow");
    }
}
```

```

if (queue->front == -1 && queue->rear == -1)
{
    queue->front = 0;
    queue->rear = 0;
    queue->queue[queue->rear] = val;
    return 0;
}

queue->rear++;
queue->queue[queue->rear] = val;
}

int dequeue(Queue *queue)
{
    if (queue->front == queue->rear)
    {
        if (queue->front == -1)
        {
            printf("Queue Underflow");
            return 0;
        }
        printf("%d\n", queue->queue[queue->front]);
        queue->front = -1;
        queue->rear = -1;
        return 0;
    }
    printf("%d\n", queue->queue[queue->front]);
    queue->front++;
}
}

int display(Queue *queue)
{
    for (int i = queue->front; i <= queue->rear; i++)
    {
        printf("%d ", queue->queue[i]);
    }
    printf("\n");
}

int main()
{

    Queue queue;
    int choice;
    int val;
    new (&queue);
    do
    {
        printf("1. For insert\n");
        printf("2. For delete\n");
        printf("3. For display\n");
        printf("4. For exit\n");

```

```

printf("Enter your choice: ");
scanf("%d", &choice);
switch (choice)
{
    case 1:
        printf("Enter value to insert: ");
        scanf("%d", &val);
        enqueue(&queue, val);
        break;
    case 2:
        dequeue(&queue);
        break;
    case 3:
        display(&queue);
        break;
    case 4:
        break;
    default:
        printf("Invalid command");
}
} while (choice != 4);
}

```

Program Code:

```

queue.c  x
queue.c > main()
You, 4 hours ago | 1 author (You)
1 #include "stdio.h"
2
3 You, 4 hours ago | 1 author (You)
4 typedef struct
5 {
6     int queue[200];
7     int front;
8     int rear;
9 } Queue;
10
11 int new (Queue *queue)
12 {
13     queue->front = -1;
14     queue->rear = -1;
15 }
16
17 int enqueue(Queue *queue, int val)
18 {
19     if (queue->front > queue->rear)
20     {
21         printf("Queue Overflow");
22     }
23     if (queue->front == -1 && queue->rear == -1)
24     {
25         queue->front = 0;
26         queue->rear = 0;
27         queue->queue[queue->rear] = val;
28     }
29     queue->rear++;
30     queue->queue[queue->rear] = val;
}

main*  0:1  0  6 hrs 14 mins
You, 4 hours ago  Ln 93, Col 27  Spaces: 4  UTF-8  LF  C  Linux

```

Activities Visual Studio Code Sun Aug 21 23:11 queue.c - lab_assessment1 - Visual Studio Code

```
queue.c  x
queue.c > main()
31 }
32
33 int dequeue(Queue *queue)
34 {
35     if (queue->front == queue->rear)
36     {
37         if (queue->front == -1)
38         {
39             printf("Queue Underflow");
40             return 0;
41         }
42         printf("%d\n", queue->queue[queue->front]);
43         queue->front = -1;
44         queue->rear = -1;
45         return 0;
46     }
47     printf("%d\n", queue->queue[queue->front]);
48     queue->front++;
49 }
50
51 int display(Queue *queue)
52 {
53     for (int i = queue->front; i <= queue->rear; i++)
54     {
55         printf("%d ", queue->queue[i]);
56     }
57     printf("\n");
58 }
59
60 int main()
61 {
62 }
```

You, 4 hours ago Ln 93, Col 27 Spaces: 4 UTF-8 LF C ⚙️ Linux ⚙️

Activities Visual Studio Code Sun Aug 21 23:11 queue.c - lab_assessment1 - Visual Studio Code

```
queue.c  x
queue.c > main()
59
60 int main()
61 {
62
63     Queue queue;
64     int choice;
65     int val;
66     new (&queue);
67     do
68     {
69         printf("1. For insert\n");
70         printf("2. For delete\n");
71         printf("3. For display\n");
72         printf("4. For exit\n");
73         printf("Enter your choice: ");
74         scanf("%d", &choice);
75         switch (choice)
76         {
77             case 1:
78                 printf("Enter value to insert: ");
79                 scanf("%d", &val);
80                 enqueue(&queue, val);
81                 break;
82             case 2:
83                 dequeue(&queue);
84                 break;
85             case 3:
86                 display(&queue);
87                 break;
88             case 4:
89                 break;
90             default:
91                 printf("Invalid command");
92         }
93     } while (choice != 4);    You, 4 hours ago + feat: added stuff
94 }
```

You, 4 hours ago Ln 93, Col 27 Spaces: 4 UTF-8 LF C ⚙️ Linux ⚙️

Activities Visual Studio Code Sun Aug 21 23:11 queue.c - lab_assessment1 - Visual Studio Code

```
queue.c  x
queue.c > main()
67     do
68     {
69         printf("1. For insert\n");
70         printf("2. For delete\n");
71         printf("3. For display\n");
72         printf("4. For exit\n");
73         printf("Enter your choice: ");
74         scanf("%d", &choice);
75         switch (choice)
76         {
77             case 1:
78                 printf("Enter value to insert: ");
79                 scanf("%d", &val);
80                 enqueue(&queue, val);
81                 break;
82             case 2:
83                 dequeue(&queue);
84                 break;
85             case 3:
86                 display(&queue);
87                 break;
88             case 4:
89                 break;
90             default:
91                 printf("Invalid command");
92         }
93     } while (choice != 4);    You, 4 hours ago + feat: added stuff
94 }
```

You, 4 hours ago Ln 93, Col 27 Spaces: 4 UTF-8 LF C ⚙️ Linux ⚙️

Output:

```
[kittycat@fedora lab_assessment1]$ gcc queue.c && ./a.out
```

1. For insert
2. For delete
3. For display
4. For exit

Enter your choice: 1

Enter value to insert: 23

1. For insert
2. For delete
3. For display
4. For exit

Enter your choice: 1

Enter value to insert: 45

1. For insert
2. For delete
3. For display
4. For exit

Enter your choice: 3

23 45

1. For insert
2. For delete
3. For display
4. For exit

Enter your choice: 2

23

1. For insert
2. For delete
3. For display
4. For exit

Enter your choice: 3

45

1. For insert
2. For delete
3. For display
4. For exit

Enter your choice: 4

```
[kittycat@fedora lab_assessment1]$
```

Output Screenshot:

The screenshot shows a terminal window in Visual Studio Code with the following content:

```
Sun Aug 21 23:16
queue.c - lab_assessment1 - Visual Studio Code
File Edit Selection View Go Run Terminal Help
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER GITLENS
bash + ×
```

The terminal output is as follows:

```
[kittycat@fedora lab_assessment1]$ gcc queue.c && ./a.out
1. For insert
2. For delete
3. For display
4. For exit
Enter your choice: 1
Enter value to insert: 23
1. For insert
2. For delete
3. For display
4. For exit
Enter your choice: 1
Enter value to insert: 45
1. For insert
2. For delete
3. For display
4. For exit
Enter your choice: 3
23 45
1. For insert
2. For delete
3. For display
4. For exit
Enter your choice: 2
23
1. For insert
2. For delete
3. For display
4. For exit
Enter your choice: 3
45
1. For insert
2. For delete
3. For display
4. For exit
Enter your choice: 4
Enter your choice: 3
```

```
Activities Visual Studio Code Sun Aug 21 23:16 queue.c - lab_assessment1 - Visual Studio Code
File Edit Selection View Go Run Terminal Help bash + ×
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER GITLENS
1. For insert
2. For delete
3. For display
4. For exit
Enter your choice: 1
Enter value to insert: 45
1. For insert
2. For delete
3. For display
4. For exit
Enter your choice: 3
23 45
1. For insert
2. For delete
3. For display
4. For exit
Enter your choice: 2
23
1. For insert
2. For delete
3. For display
4. For exit
Enter your choice: 3
45
1. For insert
2. For delete
3. For display
4. For exit
Enter your choice: 4
[KittyCat@fedora lab_assessment1]$
```

You, 4 hours ago Ln 93, Col 27 Spaces: 4 UTF-8 LF C ⌂ Linux ⌂

3. Write a program using arrays to perform insertion and deletion from both the ends of a Deque and also display the contents of it based on the choice given by the user.

Program Code:

```
#include "stdio.h"

typedef struct
{
    int deque[200];
    int front;
    int rear;
    int size;
} Deque;

int new (Deque *deque, int size)
{
    deque->front = -1;
    deque->rear = -1;
    deque->size = size;
}

int enqueue_rear(Deque *deque, int val)
{
    if ((deque->front == 0 && deque->rear == deque->size - 1) ||
        deque->front == deque->rear + 1)
    {
        printf("Deque Overflow");
    }
    if (deque->front == -1 && deque->rear == -1)
    {
        deque->front = 0;
        deque->rear = 0;
        deque->deque[deque->rear] = val;
        return 0;
    }
    else if (deque->rear == deque->size - 1 && deque->front != 0)
```

```

{
    deque->rear = 0;
    deque->deque[deque->rear] = val;
    return 0;
}

deque->rear++;
deque->deque[deque->rear] = val;
}

int enqueue_front(Deque *deque, int val)
{
    if ((deque->front == 0 && deque->rear == deque->size - 1) ||
deque->front == deque->rear + 1)
    {
        printf("Deque Overflow");
        return 0;
    }
    if (deque->front == -1 && deque->rear == -1)
    {
        deque->front = 0;
        deque->rear = 0;
        deque->deque[deque->rear] = val;
        return 0;
    }
    else if (deque->rear != deque->size - 1 && deque->front == 0)
    {
        deque->front = deque->size - 1;
        deque->deque[deque->front] = val;
        return 0;
    }
    deque->front--;
    deque->deque[deque->front] = val;
}

int deque_front(Deque *deque)
{
    if (deque->front == deque->rear)
    {
        if (deque->front == -1)
        {
            printf("Deque Underflow");
            return 0;
        }
        printf("%d\n", deque->deque[deque->front]);
        deque->front = -1;
        deque->rear = -1;
        return 0;
    }
    else if (deque->front == deque->rear - 1)
    {
        printf("%d\n", deque->deque[deque->front]);
    }
}

```

```

    deque->front = 0;
    return 0;
}
printf("%d\n", deque->deque[deque->front]);
deque->front++;
}

int deque_rear(Deque *deque)
{
    if (deque->front == deque->rear)
    {
        if (deque->front == -1)
        {
            printf("Deque Underflow");
            return 0;
        }
        printf("%d\n", deque->deque[deque->rear]);
        deque->front = -1;
        deque->rear = -1;
        return 0;
    }
    else if (deque->rear == 0)
    {
        printf("%d\n", deque->deque[deque->rear]);
        deque->rear = deque->size - 1;
        return 0;
    }
    printf("%d\n", deque->deque[deque->rear]);
    deque->rear--;
}
int display(Deque *deque){
    printf("\n");
    for(int i=deque->front;i<=deque->rear;i++){
        printf("%d ",deque->deque[i]);
    }
    printf("\n");
}
int main()
{
    Deque deque;
    int val,choice;
    new(&deque, 37);
    do
    {
        printf("1. For enqueue rear\n");
        printf("2. For enqueue front\n");
        printf("3. For deque front\n");
        printf("4. For deque rear\n");
        printf("5. For display\n");
        printf("6. For exit\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);

```

```

switch (choice)
{
    case 1:
        printf("Enter value to insert at rear: ");
        scanf("%d", &val);
        enqueue_rear(&deque, val);
        break;
    case 2:
        printf("Enter value to insert at front: ");
        scanf("%d", &val);
        enqueue_front(&deque, val);
        break;
    case 3:
        deque_front(&deque);
        break;
    case 4:
        deque_rear(&deque);
        break;
    case 5:
        display(&deque);
        break;
    case 6:
        break;
    default:
        printf("Invalid command");
}
} while (choice != 4);

}

```

Program Screenshot:

The screenshot shows the Visual Studio Code interface with the following details:

- Title Bar:** Activities, Visual Studio Code, Sun Aug 21 23:43, deque.c - lab_assessment1 - Visual Studio Code, 100%.
- File Explorer:** Shows a tree view of the project structure, including files like deque.c, deque.h, and main.c.
- Code Editor:** Displays the C code for deque.c. The code defines a Deque structure, implements enqueue and dequeue operations, and provides a menu for displaying the deque's contents.
- Bottom Status Bar:** Shows file details: deque.c, 4. For deque rear, 5. For display, main*, 0:11, 0△0, 6 hrs 24 mins, You, 4 hours ago, Ln 80, Col 6, Spaces:4, UTF-8, LF, C, Linux.

Activities Visual Studio Code Sun Aug 21 23:43 deque.c - lab_assessment1 - Visual Studio Code

```

File Edit Selection View Go Run Terminal Help
deque.c M x
C deque.c > deque_front(Deque *)
26     deque->front = 0;
27     deque->rear = 0;
28     deque->deque[deque->rear] = val;
29     return 0;
30 }
31 else if (deque->rear == deque->size - 1 && deque->front != 0)
32 {
33     deque->rear = 0;
34     deque->deque[deque->rear] = val;
35     return 0;
36 }
37 deque->rear++;
38 deque->deque[deque->rear] = val;
39 }

40 int enqueue_front(Deque *deque, int val)
41 {
42     if ((deque->front == 0 && deque->rear == deque->size - 1) || deque->front == deque->rear + 1)
43     {
44         printf("Deque Overflow");
45         return 0;
46     }
47     if (deque->front == -1 && deque->rear == -1)
48     {
49         deque->front = 0;
50         deque->rear = 0;
51         deque->deque[deque->rear] = val;
52         return 0;
53     }
54 }
55 else if (deque->rear != deque->size - 1 && deque->front == 0)
56 {
57     deque->front = deque->size - 1;
58     deque->deque[deque->front] = val;
59     return 0;
60 }
61 deque->front--;
62 deque->deque[deque->front] = val;
63 }

64 int deque_front(Deque *deque)
65 {
66     if (deque->front == deque->rear)
67     {
68         if (deque->front == -1)
69         {
70             printf("Deque Underflow");
71             return 0;
72         }
73         printf("%d\n", deque->deque[deque->front]);
74         deque->front = -1;
75         deque->rear = -1;
76         return 0;
77     }
78     else if (deque->front == deque->rear - 1)
79     {
80         You, 4 hours ago + feat: added stuff
81         printf("%d\n", deque->deque[deque->front]);
82         deque->front = 0;
83         return 0;
84     }
85     printf("%d\n", deque->deque[deque->front]);
86     deque->front++;
87 }

88 int deque_rear(Deque *deque)
89 {
90     if (deque->front == deque->rear)
91     {
92         if (deque->front == -1)
93         {
94             printf("Deque Underflow");
95             return 0;
96         }
97         printf("%d\n", deque->deque[deque->rear]);
98         deque->front = -1;
99         deque->rear = -1;
100        return 0;
101    }
102    else if (deque->rear == 0)
103    {
104        printf("%d\n", deque->deque[deque->rear]);
105        deque->rear = deque->size - 1;
106        return 0;
107    }
108 }

109

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER GITLENS

You, 4 hours ago Ln 80, Col 6 Spaces: 4 UTF-8 LF C Linux

Activities Visual Studio Code Sun Aug 21 23:44 deque.c - lab_assessment1 - Visual Studio Code

```

File Edit Selection View Go Run Terminal Help
deque.c M x
C deque.c > deque_front(Deque *)
53     return 0;
54 }
55 else if (deque->rear != deque->size - 1 && deque->front == 0)
56 {
57     deque->front = deque->size - 1;
58     deque->deque[deque->front] = val;
59     return 0;
60 }
61 deque->front--;
62 deque->deque[deque->front] = val;
63 }

64 int deque_front(Deque *deque)
65 {
66     if (deque->front == deque->rear)
67     {
68         if (deque->front == -1)
69         {
70             printf("Deque Underflow");
71             return 0;
72         }
73         printf("%d\n", deque->deque[deque->front]);
74         deque->front = -1;
75         deque->rear = -1;
76         return 0;
77     }
78     else if (deque->front == deque->rear - 1)
79     {
80         You, 4 hours ago + feat: added stuff
81         printf("%d\n", deque->deque[deque->front]);
82         deque->front = 0;
83         return 0;
84     }
85     printf("%d\n", deque->deque[deque->front]);
86     deque->front++;
87 }

88 int deque_rear(Deque *deque)
89 {
90     if (deque->front == deque->rear)
91     {
92         if (deque->front == -1)
93         {
94             printf("Deque Underflow");
95             return 0;
96         }
97         printf("%d\n", deque->deque[deque->rear]);
98         deque->front = -1;
99         deque->rear = -1;
100        return 0;
101    }
102    else if (deque->rear == 0)
103    {
104        printf("%d\n", deque->deque[deque->rear]);
105        deque->rear = deque->size - 1;
106        return 0;
107    }
108 }

109

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER GITLENS

You, 4 hours ago Ln 80, Col 6 Spaces: 4 UTF-8 LF C Linux

Activities Visual Studio Code Sun Aug 21 23:44 deque.c - lab_assessment1 - Visual Studio Code

```

File Edit Selection View Go Run Terminal Help
deque.c M x
C deque.c > deque_front(Deque *)
80     You, 4 hours ago + feat: added stuff
81     printf("%d\n", deque->deque[deque->front]);
82     deque->front = 0;
83     return 0;
84 }
85 printf("%d\n", deque->deque[deque->front]);
86 deque->front++;
87 }

88 int deque_rear(Deque *deque)
89 {
90     if (deque->front == deque->rear)
91     {
92         if (deque->front == -1)
93         {
94             printf("Deque Underflow");
95             return 0;
96         }
97         printf("%d\n", deque->deque[deque->rear]);
98         deque->front = -1;
99         deque->rear = -1;
100        return 0;
101    }
102    else if (deque->rear == 0)
103    {
104        printf("%d\n", deque->deque[deque->rear]);
105        deque->rear = deque->size - 1;
106        return 0;
107    }
108 }

109

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER GITLENS

You, 4 hours ago Ln 80, Col 6 Spaces: 4 UTF-8 LF C Linux

Activities Visual Studio Code Sun Aug 21 23:44 deque.c - lab_assessment1 - Visual Studio Code

```
File Edit Selection View Go Run Terminal Help
C deque.c M X
C deque.c > deque_front(Deque *)
108     printf("%d\n", deque->deque[deque->rear]);
109     deque->rear--;
110 }
111 int display(Deque *deque){
112     printf("\n");
113     for(int i=deque->front;i<=deque->rear;i++){
114         printf("%d ", deque->deque[i]);
115     }
116     printf("\n");
117 }
118 int main()
119 {
120     Deque deque;
121     int val,choice;
122     new(&deque, 37);
123     do
124     {
125         printf("1. For enqueue rear\n");
126         printf("2. For enqueue front\n");
127         printf("3. For deque front\n");
128         printf("4. For deque rear\n");
129         printf("5. For display\n");
130         printf("6. For exit\n");
131         printf("Enter your choice: ");
132         scanf("%d", &choice);
133         switch (choice)
134         {
135             case 1:
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER GITLENS

4. For deque rear
5. For display

main* 0:11 0 0 △ 0 6 hrs 24 mins

Sun Aug 21 23:45 deque.c - lab_assessment1 - Visual Studio Code

```
File Edit Selection View Go Run Terminal Help
C deque.c M X
C deque.c > deque_front(Deque *)
132     scanf("%d", &choice);
133     switch (choice)
134     {
135         case 1:
136             printf("Enter value to insert at rear: ");
137             scanf("%d", &val);
138             enqueue_rear(&deque, val);
139             break;
140         case 2:
141             printf("Enter value to insert at front: ");
142             scanf("%d", &val);
143             enqueue_front(&deque, val);
144             break;
145         case 3:
146             deque_front(&deque);
147             break;
148         case 4:
149             deque_rear(&deque);
150             break;
151         case 5:
152             display(&deque);
153             break;
154         case 6:
155             break;
156         default:
157             printf("Invalid command");
158     }
159 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER GITLENS

4. For deque rear
5. For display

main* 0:11 0 0 △ 0 6 hrs 24 mins

Sun Aug 21 23:45 deque.c - lab_assessment1 - Visual Studio Code

```
File Edit Selection View Go Run Terminal Help
C deque.c M X
C deque.c > deque_front(Deque *)
149     {
150         deque_rear(&deque);
151         break;
152     }
153     case 5:
154         display(&deque);
155         break;
156     case 6:
157         break;
158     default:
159         printf("Invalid command");
160     }
161 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER GITLENS

4. For deque rear
5. For display

main* 0:11 0 0 △ 0 6 hrs 24 mins

Output:

```
[kittycat@fedora lab_assessment1]$ gcc deque.c && ./a.out
```

- 1. For enqueue rear
- 2. For enqueue front
- 3. For dequeue front
- 4. For dequeue rear
- 5. For display
- 6. For exit

Enter your choice: 1

Enter value to insert at rear: 23

- 1. For enqueue rear
- 2. For enqueue front
- 3. For dequeue front
- 4. For dequeue rear
- 5. For display
- 6. For exit

Enter your choice: 1

Enter value to insert at rear: 34

- 1. For enqueue rear
- 2. For enqueue front
- 3. For dequeue front
- 4. For dequeue rear
- 5. For display
- 6. For exit

Enter your choice: 1

Enter value to insert at rear: 56

- 1. For enqueue rear
- 2. For enqueue front
- 3. For dequeue front
- 4. For dequeue rear
- 5. For display
- 6. For exit

Enter your choice: 5

23 34 56

- 1. For enqueue rear
- 2. For enqueue front
- 3. For dequeue front
- 4. For dequeue rear
- 5. For display
- 6. For exit

Enter your choice: 3

23

- 1. For enqueue rear
- 2. For enqueue front
- 3. For dequeue front
- 4. For dequeue rear
- 5. For display
- 6. For exit

Enter your choice: 5

34 56

- 1. For enqueue rear
- 2. For enqueue front
- 3. For dequeue front
- 4. For dequeue rear
- 5. For display

6. For exit

Enter your choice: 2

Enter value to insert at front: 456

1. For enqueue rear

2. For enqueue front

3. For dequeue front

4. For dequeue rear

5. For display

6. For exit

Enter your choice: 5

456 34 56

1. For enqueue rear

2. For enqueue front

3. For dequeue front

4. For dequeue rear

5. For display

6. For exit

Enter your choice: 4

56

Output Screenshot:

The screenshot shows a Visual Studio Code interface with a terminal window open. The terminal tab is selected, showing the command line and its output. The output shows the execution of a C program named 'deque.c' in a directory 'lab_assessment1'. The program performs a series of enqueue and dequeue operations on a deque, inserting values 456, 34, and 56, and then displaying the current state of the deque as 56.

```
[kittycat@fedora lab_assessment1]$ gcc deque.c && ./a.out
1. For enqueue rear
2. For enqueue front
3. For dequeue front
4. For dequeue rear
5. For display
6. For exit
Enter your choice: 1
Enter value to insert at rear: 23
1. For enqueue rear
2. For enqueue front
3. For dequeue front
4. For dequeue rear
5. For display
6. For exit
Enter your choice: 1
Enter value to insert at rear: 34
1. For enqueue rear
2. For enqueue front
3. For dequeue front
4. For dequeue rear
5. For display
6. For exit
Enter your choice: 1
Enter value to insert at rear: 56
1. For enqueue rear
2. For enqueue front
3. For dequeue front
4. For dequeue rear
5. For display
```

```

Activities  Visual Studio Code
File Edit Selection View Go Run Terminal Help
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER GITLENS
deque.c - lab_assessment1 - Visual Studio Code
Sun Aug 21 23:47
Enter value to insert at rear: 56
1. For enqueue rear
2. For enqueue front
3. For dequeue front
4. For dequeue rear
5. For display
6. For exit
Enter your choice: 5
23 34 56
1. For enqueue rear
2. For enqueue front
3. For dequeue front
4. For dequeue rear
5. For display
6. For exit
Enter your choice: 3
23
1. For enqueue rear
2. For enqueue front
3. For dequeue front
4. For dequeue rear
5. For display
6. For exit
Enter your choice: 5
34 56
1. For enqueue rear
2. For enqueue front
3. For dequeue front
4. For dequeue rear
5. For display
6. For exit
You, 4 hours ago Ln 80, Col 6 Spaces: 4 UTF-8 LF C ⚙️ Linux ⚙️
@ main* 0.11 0 △ 0 6 hrs 24 mins
Activities  Visual Studio Code
File Edit Selection View Go Run Terminal Help
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER GITLENS
deque.c - lab_assessment1 - Visual Studio Code
Sun Aug 21 23:47
6. For exit
Enter your choice: 5
34 56
1. For enqueue rear
2. For enqueue front
3. For dequeue front
4. For dequeue rear
5. For display
6. For exit
Enter your choice: 2
Enter value to insert at front: 456
1. For enqueue rear
2. For enqueue front
3. For dequeue front
4. For dequeue rear
5. For display
6. For exit
Enter your choice: 5
456 34 56
1. For enqueue rear
2. For enqueue front
3. For dequeue front
4. For dequeue rear
5. For display
6. For exit
Enter your choice: 4
56
@ [kittycat@fedora lab_assessment1]$ 
You, 4 hours ago Ln 80, Col 6 Spaces: 4 UTF-8 LF C ⚙️ Linux ⚙️
@ main* 0.11 0 △ 0 6 hrs 24 mins

```

4. Write a program that takes the details of mobile phone(model name,year,camera resolution, RAM , memory card size and Operating system)and sort the mobile phones in ascending order based on their RAM size using insertion sort.

Program Code:

```

#include "stdio.h"

typedef struct
{
    char name[200];
    int year;
    int resolution;
    int ram;
    int mem_size;
    char OS[200];
} Mobile;

int insertion_sort(Mobile *mobile, int n)
{
    int c;

```

```

Mobile m;
for (int i = 1; i < n; i++)
{
    m = *(mobile + i);
    c = i - 1;
    while (c >= 0 && m.ram < (mobile + c)->ram)
    {
        *(mobile + c + 1) = *(mobile + c);
        c--;
    }
    *(mobile + c + 1) = m;
}

int display(Mobile *mobile, int n)
{
    for (int i = 0; i < n; i++)
    {
        printf("Name:      %s\n", (mobile + i)->name);
        printf("Year:      %d\n", (mobile + i)->year);
        printf("Resolution: %d\n", (mobile + i)->resolution);
        printf("Ram:      %d\n", (mobile + i)->ram);
        printf("Memory:      %d\n", (mobile + i)->mem_size);
        printf("OS:      %s\n", (mobile + i)->OS);
    }
}

int main()
{
    Mobile mobiles[200];
    int n;
    printf("Enter number of mobiles: ");
    scanf("%d", &n);
   getc(stdin);
    for (int i = 0; i < n; i++)
    {
        printf("Enter Mobile name:");
        fgets(mobiles[i].name, 200, stdin);
        printf("Enter Mobile year:");
        scanf("%d", &mobiles[i].year);
        getc(stdin);
        printf("Enter camera resolution:");
        scanf("%d", &mobiles[i].resolution);
        getc(stdin);
        printf("Enter Ram:");
        scanf("%d", &mobiles[i].ram);
        getc(stdin);
        printf("Enter Memory card size:");
        scanf("%d", &mobiles[i].mem_size);
        getc(stdin);
        printf("Enter Mobile OS:");
    }
}

```

```

        fgets(mobiles[i].OS, 200, stdin);
        printf("\n-----\n");
    }

    insertion_sort(mobiles, n);

    display(mobiles, n);
    printf("\n");
}

```

Program Screenshot:

The screenshot shows two tabs in Visual Studio Code: 'insertionsort.c' and 'main.c'. The 'insertionsort.c' tab is active, displaying the following C code:

```

    insertion_sort(Mobile *mobile, int n)
    {
        int c;
        Mobile m;
        for (int i = 1; i < n; i++)
        {
            m = *(mobile + i);
            c = i - 1;
            while (c >= 0 && m.ram < (mobile + c)->ram)
            {
                *(mobile + c + 1) = *(mobile + c);
                c--;
            }
            *(mobile + c + 1) = m;
        }
    }

    int display(Mobile *mobile, int n)

```

The 'main.c' tab shows the following C code:

```

    int main()
    {
        Mobile mobiles[200];
        int n;
        printf("Enter number of mobiles: ");
        scanf("%d", &n);
        get(stdin);
        for (int i = 0; i < n; i++)
        {
            printf("Enter Mobile name:");
            fgets(mobiles[i].name, 200, stdin);
            printf("Enter Mobile year:");
            scanf("%d", &mobiles[i].year);
            get(stdin);
            printf("Enter camera resolution:");
            scanf("%d", &mobiles[i].resolution);
            get(stdin);
            printf("Enter Ram:");
            scanf("%d", &mobiles[i].ram);
            get(stdin);
        }
    }

```

A screenshot of the Visual Studio Code interface. The title bar shows "Activities" and "Visual Studio Code" at the top left, and "Mon Aug 22 00:26" and "insertionsort.c - lab_assessment1 - Visual Studio Code" at the top right. The status bar at the bottom right shows "You, 3 days ago Ln 14, Col 2 Spaces: 4 UTF-8 LF C ⚙️ Linux". The main editor area contains the following C code:

```
insertionsort.c
1  insertion sort(Mobile * , int);
2
3  insertion sort > 56
4      Scanf("%d", &mobiles[i].year);
5      getc(stdin);
6      printf("Enter camera resolution:");
7      scanf("%d", &mobiles[i].resolution);
8      getc(stdin);
9      printf("Enter Ram:");
10     scanf("%d", &mobiles[i].ram);
11     getc(stdin);
12     printf("Enter Memory card size:");
13     scanf("%d", &mobiles[i].mem_size);
14     getc(stdin);
15     printf("Enter Mobile OS:");
16     fgets(mobiles[i].OS, 200, stdin);
17     printf("\n-----\n");
18
19 }
20
21 insertion_sort(mobiles, n);
22
23 display(mobiles, n);
24 printf("\n");
25 }
```

Output:

```
[kittycat@fedora lab_assessment1]$ gcc insertionsort.c && ./a.out
```

```
Enter number of mobiles: 4
```

```
Enter Mobile name:Samsung M40
```

```
Enter Mobile year:2021
```

```
Enter camera resolution:45
```

```
Enter Ram:8
```

```
Enter Memory card size:128
```

```
Enter Mobile OS:Android 10
```

```
-----  
Enter Mobile name:Apple 10
```

```
Enter Mobile year:2020
```

```
Enter camera resolution:34
```

```
Enter Ram:6
```

```
Enter Memory card size:64
```

```
Enter Mobile OS:IOS
```

```
-----  
Enter Mobile name:Samsung M20
```

```
Enter Mobile year:2019
```

```
Enter camera resolution:20
```

```
Enter Ram:4
```

```
Enter Memory card size:64
```

```
Enter Mobile OS:Android 9
```

```
-----  
Enter Mobile name:OPPO
```

```
Enter Mobile year:2019
```

```
Enter camera resolution:23
```

```
Enter Ram:2
```

```
Enter Memory card size:32
```

```
Enter Mobile OS:Android
```

```
Name: OPPO
```

```
Year: 2019
```

Resolution: 23

Ram: 2

Memory: 32

OS: Android

Name: Samsung M20

Year: 2019

Resolution: 20

Ram: 4

Memory: 64

OS: Android 9

Name: Apple 10

Year: 2020

Resolution: 34

Ram: 6

Memory: 64

OS: IOS

Name: Samsung M40

Year: 2021

Resolution: 45

Ram: 8

Memory: 128

OS: Android 10

[kittycat@fedora lab_assessment1]\$

Output Screenshot:

The screenshot shows the Visual Studio Code interface with the terminal tab active. The terminal window displays the following command-line session:

```
[kittycat@fedora lab_assessment1]$ gcc insertionsort.c && ./a.out
Enter number of mobiles: 4
Enter Mobile name:Samsung M40
Enter Mobile year:2021
Enter camera resolution:45
Enter Ram:8
Enter Memory card size:128
Enter Mobile OS:Android 10

-----
Enter Mobile name:Apple 10
Enter Mobile year:2020
Enter camera resolution:34
Enter Ram:6
Enter Memory card size:64
Enter Mobile OS:IOS

-----
Enter Mobile name:Samsung M20
Enter Mobile year:2019
Enter camera resolution:20
Enter Ram:4
Enter Memory card size:64
Enter Mobile OS:Android 9

-----
Enter Mobile name:OPPO
Enter Mobile year:2019
Enter camera resolution:23
Enter Ram:2
```

The Explorer sidebar shows a folder named "LAB_ASSESSMENT1" containing several C source files: editorconfig, a.out, balance.py, brackets.c, deque.c, infix_postfix.c, infix_rpn.c, insertionsort.c, patient.c, queue.c, and rpn.c. There is also a file named VL202223010381.. and a folder icon.

```

Activities  Visual Studio Code
File Edit Selection View Go Run Terminal Help
insertionsort.c - lab_assessment1 - Visual Studio Code
Sun Aug 21 23:58
TERMINAL JUPYTER GITLENS
bash + ×

U .editorconfig Enter Memory card size:64
M a.out Enter Mobile OS:Android 9
balance.py
brackets.c
deque.c Enter Mobile name:OPPO
infix_postfix.c Enter Mobile year:2019
infix_rpn.c Enter camera resolution:23
insertionsort.c Enter Ram:2
patient.c Enter Memory card size:32
queue.c Enter Mobile OS:Android
rpn.c
VL202223010381... Name: OPPO

Year: 2019
Resolution: 23
Ram: 2
Memory: 32
OS: Android

Name: Samsung M20

Year: 2019
Resolution: 20
Ram: 4
Memory: 64
OS: Android 9

Name: Apple 10

You, 3 days ago Ln 14, Col 2 Spaces: 4 UTF-8 LF C ⚙️ Linux ⚙️

```



```

Activities  Visual Studio Code
File Edit Selection View Go Run Terminal Help
insertionsort.c - lab_assessment1 - Visual Studio Code
Sun Aug 21 23:58
TERMINAL JUPYTER GITLENS
bash + ×

U .editorconfig Ram: 2
M a.out Memory: 32
balance.py OS: Android
brackets.c
deque.c Name: Samsung M20
infix_postfix.c
infix_rpn.c Year: 2019
insertionsort.c Resolution: 20
patient.c Ram: 4
queue.c Memory: 64
rpn.c OS: Android 9
VL202223010381... Name: Apple 10

Year: 2020
Resolution: 34
Ram: 6
Memory: 64
OS: IOS

Name: Samsung M40

Year: 2021
Resolution: 45
Ram: 8
Memory: 128
OS: Android 10

You, 3 days ago Ln 14, Col 2 Spaces: 4 UTF-8 LF C ⚙️ Linux ⚙️

```

5. Write a program that takes the details of a patient(hospital number, patient name, age, token number, height, , weight, reason(disease) and sort the patients in ascending order based on their token number using quick sort.

Program Code:

```
#include <stdio.h>

typedef struct{
    char name[200];
    int age;
    int token_number;
    int height;
    int weight;
    char reason[200];
} Patient;
```

```
int quicksort(Patient *patient,int f,int l){
    int i=0,j=0,pivot=0;
    Patient temp;
    if (f<l)
```

```

{
    pivot=f;
    i=f;
    j=l;

    while (i<j)
    {
        while ((patient+i)->token_number<=(patient+pivot)->token_number
&& i<l)
        {
            i++;
        }
        while ((patient+j)->token_number>=(patient+pivot)->token_number
&& j>f)
        {
            j--;
        }
        if(i<j){
            temp=*(patient+i);
            *(patient+i)=*(patient+j);
            *(patient+j)=temp;
        }
        temp=*(patient+j);
        *(patient+j)=*(patient+pivot);
        *(patient+pivot)=temp;

        quicksort(patient,f,j-1);
        quicksort(patient,j+1,l);
    }
}

void disp(Patient *patient,int n){
    for (size_t i = 0; i < n; i++)
    {
        printf("\n-----\n",i+1);
        printf("Name : %s \n", (patient+i)->name);
        printf("Age : %d \n", (patient+i)->age);
        printf("Token Number : %d \n", (patient+i)->token_number);
        printf("Height : %d \n", (patient+i)->height);
        printf("Weight : %d \n", (patient+i)->weight);
        printf("Reason : %s \n", (patient+i)->reason);
    }
    printf("\n");
}

int main(){
    int n;
    Patient patient[200];
    printf("Enter the number of patients: ");
    scanf("%d",&n);
    getchar();
    for (int i = 0; i < n; i++)

```

```

{
    printf("\n-----Patient: %d-----\n", i+1);
    printf("Enter name: ");
    fgets(patient[i].name, 200, stdin);
    printf("Enter age: ");
    scanf("%d", &patient[i].age);
    getchar();
    printf("Enter token number: ");
    scanf("%d", &patient[i].token_number);
    getchar();
    printf("Enter height: ");
    scanf("%d", &patient[i].height);
    getchar();
    printf("Enter weight: ");
    scanf("%d", &patient[i].weight);
    getchar();
    printf("Enter reason: ");
    fgets(patient[i].reason, 200, stdin);

}

quicksort(patient, 0, n-1);
disp(patient, n);
}

```

Program Screenshot:

The screenshot shows the Visual Studio Code interface with the following details:

- Title Bar:** Activities - Visual Studio Code, Mon Aug 22 00:02, patient.c - lab_assessment1 - Visual Studio Code
- File Explorer:** Shows files: 'dequeue.c M' and 'patient.c X'. The 'patient.c' file is selected.
- Code Editor:** Displays the C code for the quicksort algorithm. The code defines a 'Patient' struct and implements the quicksort function using a pivot-based partitioning scheme.
- Status Bar:** Shows file statistics: 0 main*, 0:11, 0 △ 0, and a terminal icon.

```

#include "stdio.h"

typedef struct{
    char name[200];
    int age;
    int token_number;
    int height;
    int weight;
    char reason[200];
} Patient;

int quicksort(Patient *patient, int f, int l){
    int i=0, j=0, pivot=0;
    Patient temp;
    if (f<l)
    {
        pivot=f;
        i=f;
        j=l;
        while (i<j)
        {
            while ((patient+i)->token_number<=(patient+pivot)->token_number && i<l)
            {
                i++;
            }
            while ((patient+j)->token_number>=(patient+pivot)->token_number && j>f)
            {
                j--;
            }
            if (i<=j)
            {
                temp=*(patient+i);
                *(patient+i)=*(patient+j);
                *(patient+j)=temp;
                i++;
                j--;
            }
        }
    }
    return 0;
}

```

File Edit Selection View Go Run Terminal Help

Mon Aug 22 00:02 patient.c - lab_assessment1 - Visual Studio Code

```

C deque.c M C patient.c X
C patient.c > quicksort(Patient *, int, int)
31     if(i<j){
32         temp=*(patient+i);
33         *(patient+i)=*(patient+j);
34         *(patient+j)=temp;
35     }
36 }
37 temp=*(patient+j);
38 *(patient+j)=*(patient+pivot);
39 *(patient+pivot)=temp;
40
41 quicksort(patient,f,j-1);
42 quicksort(patient,j+1,l);
43 }      You, 4 hours ago * feat: added stuff
44 }
45 void disp(Patient *patient,int n){
46     for (size_t i = 0; i < n; i++)
47     {
48         printf("\n-----\n",i+1);
49         printf("Name : %s \n", (patient+i)->name);
50         printf("Age : %d \n", (patient+i)->age);
51         printf("Token Number : %d \n", (patient+i)->token_number);
52         printf("Height : %d \n", (patient+i)->height);
53         printf("Weight : %d \n", (patient+i)->weight);
54         printf("Reason : %s \n", (patient+i)->reason);
55     }
56     printf("\n");
57 }
58 int main(){
59     int n;
60     Patient patient[200];
61     printf("Enter the number of patients: ");
62     scanf("%d",&n);
63
64     main* 0.11 0 0 △ 0

```

You, 4 hours ago Ln 43, Col 6 Spaces: 4 UTF-8 LF C 🔍 Linux ⌂ ⌂ ⌂

File Edit Selection View Go Run Terminal Help

Mon Aug 22 00:03 patient.c - lab_assessment1 - Visual Studio Code

```

C deque.c M C patient.c X
C patient.c > quicksort(Patient *, int, int)
60 Patient patient[200];
61 printf("Enter the number of patients: ");
62 scanf("%d",&n);
63 getchar();
64 for (int i = 0; i < n; i++)
65 {
66     printf("\n-----Patient: %d-----\n",i+1);
67     printf("Enter name: ");
68     fgets(patient[i].name,200,stdin);
69     printf("Enter age: ");
70     scanf("%d",&patient[i].age);
71     getchar();
72     printf("Enter token number: ");
73     scanf("%d",&patient[i].token_number);
74     getchar();
75     printf("Enter height: ");
76     scanf("%d",&patient[i].height);
77     getchar();
78     printf("Enter weight: ");
79     scanf("%d",&patient[i].weight);
80     getchar();
81     printf("Enter reason: ");
82     fgets(patient[i].reason,200,stdin);
83
84 }
85
86 quicksort(patient,0,n-1);
87 disp(patient,n);
88 }

main* 0.11 0 0 △ 0

```

You, 4 hours ago Ln 43, Col 6 Spaces: 4 UTF-8 LF C 🔍 Linux ⌂ ⌂ ⌂

Program Output:

[kittycat@fedora lab_assessment1]\$ gcc patient.c && ./a.out

Enter the number of patients: 4

-----Patient: 1-----

Enter name: Utkarsh

Enter age: 20

Enter token number: 45

Enter height: 180

Enter weight: 89

Enter reason: Obesity

-----Patient: 2-----

Enter name: Ujjawal

Enter age: 19

Enter token number: 25

Enter height: 175

Enter weight: 67

Enter reason: Depression

-----Patient: 3-----

Enter name: Tejas
Enter age: 18
Enter token number: 12
Enter height: 165
Enter weight: 56
Enter reason: Insomnia

-----Patient: 4-----

Enter name: Murali
Enter age: 45
Enter token number: 34
Enter height: 125
Enter weight: 67
Enter reason: Obesity

Name : Tejas

Age : 18
Token Number : 12
Height : 165
Weight : 56
Reason : Insomnia

Name : Ujjawal

Age : 19
Token Number : 25
Height : 175
Weight : 67
Reason : Depression

Name : Murali

Age : 45
Token Number : 34
Height : 125
Weight : 67
Reason : Obesity

Name : Utkarsh

Age : 20
Token Number : 45
Height : 180
Weight : 89
Reason : Obesity

[kittycat@fedora lab_assessment1]\$

Output Screenshot:

Activities Visual Studio Code

File Edit Selection View Go Run Terminal Help

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER GITLENS

[kittycat@fedora lab_assessment1]\$ gcc patient.c && ./a.out

Enter the number of patients: 4

-----Patient: 1-----
Enter name: Utkarsh
Enter age: 20
Enter token number: 45
Enter height: 180
Enter weight: 89
Enter reason: Obesity

-----Patient: 2-----
Enter name: Ujjawal
Enter age: 19
Enter token number: 25
Enter height: 175
Enter weight: 67
Enter reason: Depression

-----Patient: 3-----
Enter name: Tejas
Enter age: 18
Enter token number: 12
Enter height: 165
Enter weight: 56
Enter reason: Insomnia

-----Patient: 4-----
Enter name: Murali
Enter age: 45

You, 4 hours ago Ln 52, Col 54 Spaces:4 UTF-8 LF C Linux

Activities Visual Studio Code

File Edit Selection View Go Run Terminal Help

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER GITLENS

[kittycat@fedora lab_assessment1]\$ gcc patient.c && ./a.out

-----Patient: 4-----
Enter name: Murali
Enter age: 45
Enter token number: 34
Enter height: 125
Enter weight: 67
Enter reason: Obesity

Name : Tejas

Name : Ujjawal

Age : 18
Token Number : 12
Height : 165
Weight : 56
Reason : Insomnia

Name : Ujjawal

Age : 19
Token Number : 25
Height : 175
Weight : 67
Reason : Depression

Name : Murali

You, 4 hours ago Ln 52, Col 54 Spaces:4 UTF-8 LF C Linux

Activities Visual Studio Code

File Edit Selection View Go Run Terminal Help

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER GITLENS

[kittycat@fedora lab_assessment1]\$ gcc patient.c && ./a.out

Name : Ujjawal

Age : 19
Token Number : 25
Height : 175
Weight : 67
Reason : Depression

Name : Murali

Age : 45
Token Number : 34
Height : 125
Weight : 67
Reason : Obesity

Name : Utkarsh

Age : 20
Token Number : 45
Height : 180
Weight : 89
Reason : Obesity

[kittycat@fedora lab_assessment1]\$

You, 4 hours ago Ln 52, Col 54 Spaces:4 UTF-8 LF C Linux