

Name: Kulvir Singh

Register Number:19BCE2074

Q1>Check for Palindrome word using stack

The screenshot shows the OnlineGDB beta compiler interface. The code in the editor is as follows:

```
#include<stdio.h>
#include<stdlib.h>
#define max 20
char ch[20],ch1[20];
int top=-1;
int n,k=0;
void create()
{
    printf("enter the size of the word to be entered\n");
    scanf("%d",&n);
}
void push()
{
    if(top==n-1)
        printf("max length exceeded\n");
    else
    {
        top++;
        scanf("%s",&ch[top]);
    }
}
void pop()
{
    if(top==-1)
    {
        printf("entire word emptied\n");
    }
}
```

The interface includes a sidebar with user information (Welcome, kulvir06), project creation options, and social sharing links. The status bar at the bottom shows system information like battery level, signal strength, and date/time.

The screenshot shows the OnlineGDB beta compiler interface with the complete C code for checking a palindrome using a stack. The code includes the implementation of the stack operations (push, pop) and the logic to check if the word is a palindrome by comparing characters from both ends.

```
void pop()
{
    if(top== -1)
    {
        printf("entire word emptied\n");
    }
    else
    {
        ch1[k]=ch[top];
        top--;
        k++;
    }
}
void check()
{
    int i,flag=0;
    for(i=0;i<k;i++)
    if(ch[i]!=ch1[i])
    {
        flag=1;
        break;
    }
    if(flag==1)
        printf("not a pallindrome word\n");
    else
        printf("pallindrome word\n");
}
```

The interface remains consistent with the previous screenshot, showing the same sidebar, social sharing, and system status bar.

onlinegdb.com/online_c_compiler

OnlineGDB beta

Welcome, **kulvir06**

Create New Project

My Projects

Learn Programming

Programming Questions

Jobs new

Logout

f | Twitter | + 32.2K

About • FAQ • Blog • Terms of Use • Contact
Us • GDB Tutorial • Credits • Privacy
© 2016 - 2019 GDB Online

main.c

```
34 }
35 void check()
36 {
37     int i,flag=0;
38     for(i=0;i<k;i++)
39     if(ch[i]!=ch1[i])
40     {
41         flag=1;
42         break;
43     }
44     if(flag==1)
45     printf("not a pallindrome word\n");
46     else
47     printf("pallindrome word\n");
48 }
49 void main()
50 {
51     create();
52     int i;
53     printf("enter the word letter-wise\n");
54     for(i=0;i<n;i++)
55     push();
56     for(i=0;i<n;i++)
57     pop();
58     check();
59 }
60 }
```

input

ENG 4:11 PM IN 12/19/2019

onlinegdb.com/online_c_compiler

OnlineGDB beta

Welcome, **kulvir06**

Create New Project

My Projects

Learn Programming

Programming Questions

Jobs new

Logout

f | Twitter | + 32.2K

About • FAQ • Blog • Terms of Use • Contact
Us • GDB Tutorial • Credits • Privacy
© 2016 - 2019 GDB Online

main.c

```
enter the size of the word to be entered
5
enter the word letter-wise
a
r
o
r
a
a
pallindrome word

...Program finished with exit code 17
Press ENTER to exit console.
```

input

ENG 4:12 PM IN 12/19/2019

```

enter the size of the word to be entered
4
enter the word letter-wise
a
b
c
d
not a pallindrome word

...Program finished with exit code 23
Press ENTER to exit console.

```

Experiments on Queues

1>The CPU handles the requests from various devices connected to the system like keyboard, mouse, scanner, printer etc., Whenever a new request arrives, it has to check whether the queue is full or not similarly whenever you have to delete a request it has to check whether the queue is empty or not.
Emulate such an environment in C using suitable data structure and explain how the requests are handled. Perform the following operations on a queue of size 5. 1. dequeue(), 2. enqueue(), 3.enqueue(), 4.dequeue(), 5.display(), 6.dequeue(), 7.dequeue(), 8.display()

```

main.c
1 #include<stdio.h>
2 #include<stdlib.h>
3 #define m 5
4 int q[m];int front=-1;int rear=-1;
5 void isFull()
6 {
7     if(front===-1 && rear===-1)
8         printf("Queue is empty\n");
9     else
10        printf("Queue is unempty\n");
11 }
12 void enqueue()
13 {
14     if(front===-1 && rear===-1)
15     {
16         front++;rear++;
17         printf("enter element in queue \n");
18         scanf("%d",&q[rear]);
19     }
20     else if(front==4||rear==4)
21         printf("Queue overflow\n");
22     else
23     {
24         rear++;
25         printf("enter element in queue \n");
26         scanf("%d",&q[rear]);
27     }
28 }

```

VL2019205000932_AST01.pdf

Online C Compiler - online editor

C Preprocessor and Macros

segmentation fault in c - Google

onlinegdb.com/online_c_compiler

Apps fat question bank Array Of C String ... Online C Compiler ...

Language C

OnlineGDB beta

online compiler and debugger for c/c++

code. compile. run. debug. share.

IDE

My Projects

Learn Programming

Programming Questions

Jobs new

Sign Up

Login

f 32.1K

About • FAQ • Blog • Terms of Use • Contact

Us • GDB Tutorial • Credits • Privacy

© 2016 - 2019 GDB Online

main.c

```
27 }
28 }
29 void dequeue()
30 {
31     if(front===-1&&rear===-1)
32         printf("queue underflow\n");
33     else if(front>rear)
34         printf("queue underflow\n");
35     else
36     {
37         printf("deleted element %d\n",q[front]);
38         front++;
39     }
40 }
41 void display()
42 {
43     if(front===-1&&rear===-1)
44         printf("queue empty\n");
45     else
46     {
47         int i;
48         printf("Queue : \n");
49         int x=check();
50         if(x==0)
51             printf("All elements deleted from queue\n");
52         else{
53             for(i=front;i<=rear;i++)
54                 printf("%d\n",q[i]);
55             printf("\n");
56         }
57     }
58 }
59 int check()
60 {
61     int c=0,i;
62     for(i=front;i<=rear;i++)
63     {
64         if(q[i]!=NULL)
65             c++;
66     }
67 }
68 return c;
69 }
70 }
71 int main()
72 {
```

Windows taskbar:

- Type here to search
- File Explorer
- Google Chrome
- File Manager
- OneDrive
- PowerShell
- File History
- Task View
- System
- Network
- Power
- Volume
- ENG 1055 PM IN 12/17/2019

VL2019205000932_AST01.pdf

Online C Compiler - online editor

C Preprocessor and Macros

segmentation fault in c - Google

onlinegdb.com/online_c_compiler

Apps fat question bank Array Of C String ... Online C Compiler ...

Language C

OnlineGDB beta

online compiler and debugger for c/c++

code. compile. run. debug. share.

IDE

My Projects

Learn Programming

Programming Questions

Jobs new

Sign Up

Login

f 32.1K

About • FAQ • Blog • Terms of Use • Contact

Us • GDB Tutorial • Credits • Privacy

© 2016 - 2019 GDB Online

main.c

```
46 {
47     int i;
48     printf("Queue : \n");
49     int x=check();
50     if(x==0)
51         printf("All elements deleted from queue\n");
52     else{
53         for(i=front;i<=rear;i++)
54             printf("%d\n",q[i]);
55             printf("\n");
56     }
57 }
58 int check()
59 {
60     int c=0,i;
61     for(i=front;i<=rear;i++)
62     {
63         if(q[i]!=NULL)
64             c++;
65     }
66 }
67 return c;
68 }
69 }
70 }
71 int main()
72 {
```

Windows taskbar:

- Type here to search
- File Explorer
- Google Chrome
- File Manager
- OneDrive
- PowerShell
- File History
- Task View
- System
- Network
- Power
- Volume
- ENG 1055 PM IN 12/17/2019

The screenshot shows the OnlineGDB beta IDE interface. On the left, there's a sidebar with links like 'My Projects', 'Learn Programming', 'Programming Questions', 'Jobs', 'Sign Up', and 'Login'. The main area displays a C code editor with the following code:

```
main.c
56     }
57 }
58 int check()
59 {
60     int c=0,i;
61     for(i=front;i<=rear;i++)
62     {
63         if(q[i]==NULL)
64             c++;
65     }
66     return c;
67 }
68 int main()
69 {
70     dequeue();
71     enqueue();
72     enqueue();
73     dequeue();
74     display();
75     dequeue();
76     display();
77     dequeue();
78     display();
79 }
```

The status bar at the bottom shows the date and time: © 2016 - 2019 GDB Online, ENG 10:55 PM IN 12/17/2019.

The screenshot shows the OnlineGDB IDE interface with the same sidebar and code editor as the first one. The main area now displays the output of the program:

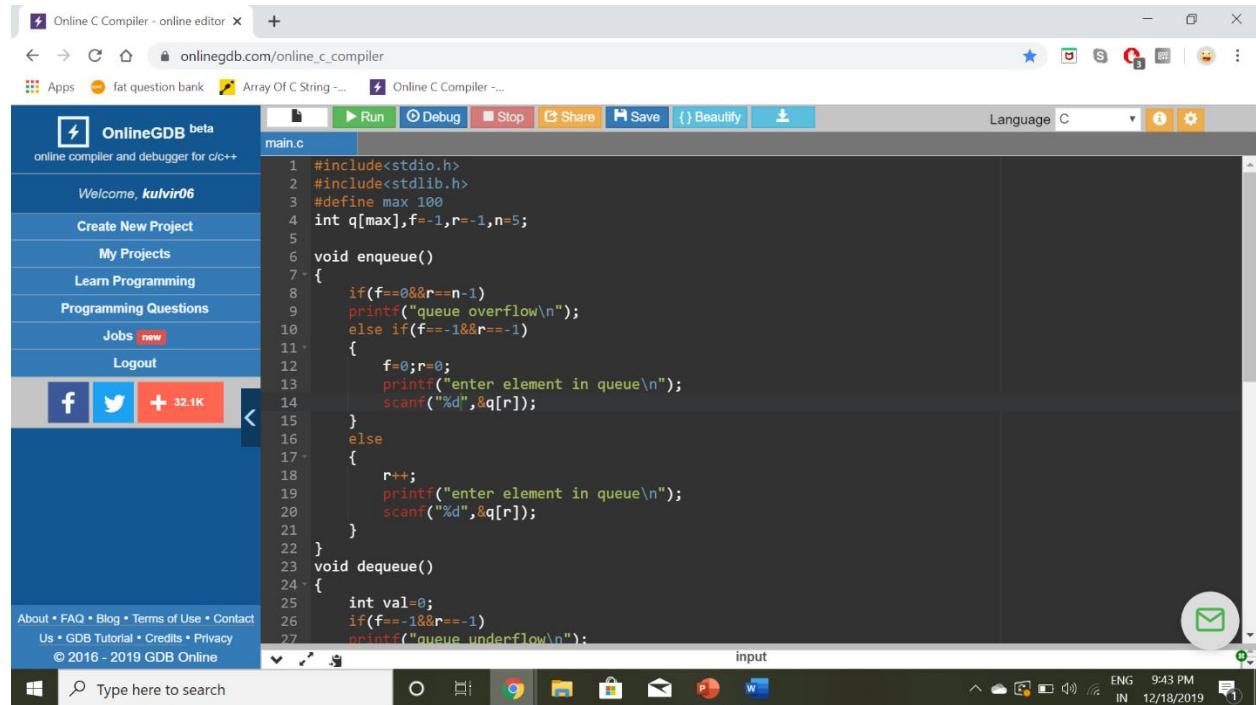
```
input
main.c:49:15: warning: implicit declaration of function 'check' [-Wimplicit-function-declaration]
main.c:65:16: warning: comparison between pointer and integer
queue underflow
enter element in queue
1234
enter element in queue
564
deleted element 1234
Queue :
564
deleted element 564
queue underflow
Queue :
All elements deleted from queue

...Program finished with exit code 0
Press ENTER to exit console.
```

The status bar at the bottom shows the date and time: © 2016 - 2019 GDB Online, ENG 10:56 PM IN 12/17/2019.

2>. The CPU handles the requests from various devices connected to the system like keyboard, mouse, scanner, printer etc., Whenever a new request arrives, it has to check whether the queue is full or not similarly whenever you have to delete a request it has to check whether the queue is empty or not. Emulate such an environment in C using suitable data structure and explain how the requests are handled. Perform the following operations on a circular queue of size 5. 1. Enqueue() 2. Enqueue() 3.

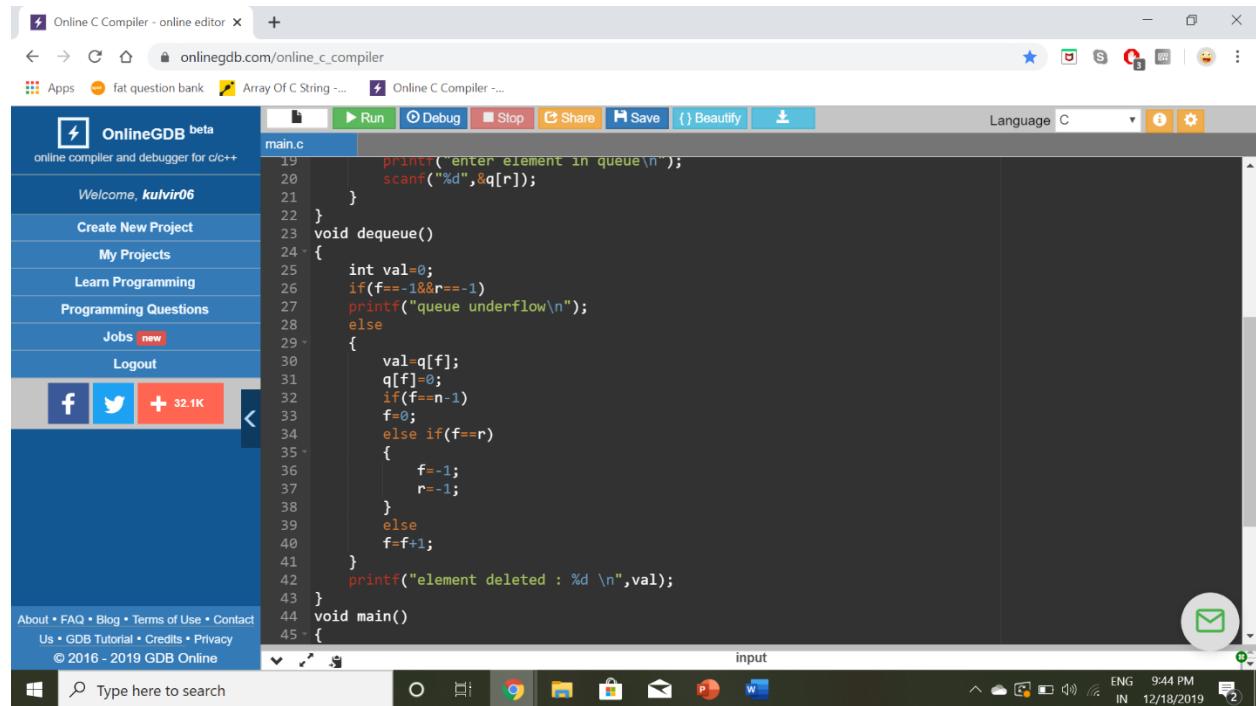
Enqueue() 4. Enqueue() 5. Enqueue() 6. Enqueue() 7. Dequeue() 8. Enqueue() 9. Enqueue() 10. Dequeue() 11. Enqueue()



The screenshot shows the OnlineGDB beta compiler interface. The code editor contains the following C code:

```
#include<stdio.h>
#include<stdlib.h>
#define max 100
int q[max],f=-1,r=-1,n=5;
void enqueue()
{
    if(f==0&&r==n-1)
        printf("queue overflow\n");
    else if(f==-1&&r==-1)
    {
        f=0;r=0;
        printf("enter element in queue\n");
        scanf("%d",&q[r]);
    }
    else
    {
        r++;
        printf("enter element in queue\n");
        scanf("%d",&q[r]);
    }
}
void dequeue()
{
    int val=0;
    if(f==-1&&r==-1)
        printf("queue underflow\n");
}
```

The status bar at the bottom right shows: ENG 9:43 PM IN 12/18/2019.



The screenshot shows the OnlineGDB beta compiler interface. The code editor contains the following C code:

```
main.c
19     printf(" enter element in queue\n");
20     scanf("%d",&q[r]);
21 }
22 void dequeue()
23 {
    int val=0;
24     if(f==-1&&r==-1)
25         printf("queue underflow\n");
26     else
27     {
28         val=q[f];
29         q[f]=0;
30         if(f==n-1)
31             f=0;
32         else if(f==r)
33         {
34             f=-1;
35             r=-1;
36         }
37         else
38             f=f+1;
39     }
40     printf("element deleted : %d \n",val);
41 }
42 void main()
43 {
44 }
```

The status bar at the bottom right shows: ENG 9:44 PM IN 12/18/2019.

The screenshot shows a web-based online compiler environment. The URL is onlinegdb.com/online_c_compiler. The main workspace displays a C program named `main.c` with line numbers. The code implements a queue using arrays and pointers. It includes enqueue and dequeue operations. The interface features a toolbar with buttons for Run, Debug, Stop, Share, Save, and Beautify. A sidebar on the left provides navigation links like Welcome, Create New Project, My Projects, Learn Programming, Programming Questions, Jobs, and Logout, along with social sharing icons for Facebook, Twitter, and LinkedIn. The bottom of the screen shows the Windows taskbar with various pinned icons.

```
int f=-1;
int r=-1;
if(f==r)
{
    f=-1;
    r=-1;
}
else
    f=f+1;
printf("element deleted : %d \n",val);
}
void main()
{
enqueue();
enqueue();
enqueue();
enqueue();
enqueue();
enqueue();
dequeue();
enqueue();
enqueue();
dequeue();
enqueue();
}
```

The screenshot shows a web-based C/C++ compiler interface. The left sidebar has links for 'Welcome, kulvir06', 'Create New Project', 'My Projects', 'Learn Programming', 'Programming Questions', 'Jobs new', and 'Logout'. It also features social sharing buttons for Facebook, Twitter, and a red button with '+ 32.1K'. The main area displays a terminal window with the following text:

```
enter element in queue
1
enter element in queue
2
enter element in queue
3
enter element in queue
4
enter element in queue
5
queue overflow
element deleted : 1
enter element in queue
6
enter element in queue
7
element deleted : 2
enter element in queue
8

...Program finished with exit code 1
Press ENTER to exit console.
```

At the bottom, there's a search bar and system status icons for battery, signal, and volume.

Extra Questions:

1>Create a data structure that efficiently supports the stack operations (pop and push) and also returns the maximum element. Assume the elements are integers or float so that you can compare them. Hint: use two stacks, one to store all of the elements and a second stack to store the maximums.

The screenshot shows the OnlineGDB interface. The code in the editor is as follows:

```
#include<stdio.h>
#include<stdlib.h>
#define m 100
int top=-1;
int s1[m],s2[m],n,k=0,max=0;
void create()
{
    printf("enter the size of the stack\n");
    scanf("%d",&n);
}
void push()
{
    if(top==n-1)
        printf("stack overflow\n");
    else
    {
        top++;
        printf("enter element in the stack\n");
        scanf("%d",&s1[top]);
    }
}
void pop()
{
    if(top==-1)
        printf("stack underflow\n");
    else
    {
        printf("removing element from the stack leads to a new maximum element\n");
        top--;
    }
}
```

The screenshot shows the OnlineGDB interface with the completed C code. The code now includes a `maxx()` function to calculate the maximum value in the stack.

```
#include<stdio.h>
#include<stdlib.h>
#define m 100
int top=-1;
int s1[m],s2[m],n,k=0,max=0;
void create()
{
    printf("enter the size of the stack\n");
    scanf("%d",&n);
}
void push()
{
    if(top==n-1)
        printf("stack overflow\n");
    else
    {
        top++;
        printf("enter element in the stack\n");
        scanf("%d",&s1[top]);
    }
}
void pop()
{
    if(top==-1)
        printf("stack underflow\n");
    else
    {
        printf("removing element from the stack leads to a new maximum element\n");
        top--;
    }
}
void maxx()
{
    int i;
    for(i=0;i<top;i++)
    {
        if(s1[i]>max)
            max=s1[i];
    }
    s2[k]=max;
    k++;
}
display maxx()
{
    printf("max elements array :\n");
    int i;
    for(i=0;i<k;i++)
    {
        printf("%d \t",s2[i]);
        printf("\n");
    }
}
void main()
{
    create();
    push();
}
```

The screenshot shows the OnlineGDB beta interface. On the left, there's a sidebar with navigation links like 'Welcome, kulvir06', 'DSA ASSIGNMENT 1', 'Create New Project', 'My Projects', 'Learn Programming', 'Programming Questions', 'Jobs new', and 'Logout'. Below the sidebar are social media sharing icons for Facebook, Twitter, and LinkedIn, and a '52.9K' link. The main area displays a C code editor with the following code:

```
main.c
38     max=s1[i];
39 }
40 s2[k]=max;
41 k++;
42 }
43 void display_maxx()
44 {
45     printf("max elements array :\n");
46     int i;
47     for(i=0;i<k;i++)
48         printf("%d \t",s2[i]);
49     printf("\n");
50 }
51 void main()
52 {
53     create();
54     push();
55     push();
56     push();
57     maxx();
58     display_maxx();
59     pop();
60     maxx();
61     display_maxx();
62     push();
63     maxx();
64     display_maxx();
65 }
```

The status bar at the bottom indicates 'ENG 3:17 PM IN 12/18/2019'.

The screenshot shows the execution results of the C program. The terminal window displays the following output:

```
main.c:44:1: warning: return type defaults to 'int' [-Wimplicit-int]
enter the size of the stack
3
enter element in the stack
123
enter element in the stack
4332
enter element in the stack
431
max elements array :
4332
removing element from the stack leads to a new maximum element
max elements array :
4332    4332
enter element in the stack
9999
max elements array :
4332    4332    9999

...Program finished with exit code 10
Press ENTER to exit console.
```

The status bar at the bottom indicates 'ENG 3:17 PM IN 12/18/2019'.

5>Write a C/C++ program to eliminate duplicates in a stack, including the original value. For e.g., 2 3 4 1
1 4 7 8 2 reduces to 3 7 8

The screenshot shows the OnlineGDB beta interface. The left sidebar includes links for Welcome, DSA ASSIGNMENT 1, Create New Project, My Projects, Learn Programming, Programming Questions, Jobs, Logout, and social media links for Facebook, Twitter, and LinkedIn. The main area displays a C program named 'main.c' with the following code:

```
1 #include<stdio.h>
2 #include<stdlib.h>
3 #define max 100
4 int stk1[max],stk2[max],top=-1,n,k=0;
5 void create()
6 {
7     printf("enter size of stack\n");
8     scanf("%d",&n);
9 }
10 void push()
11 {
12     if(top==n-1)
13         printf("stack overflow\n ");
14     else
15     {
16         top++;
17         printf("enter element in stack\n");
18         scanf("%d",&stk1[top]);
19     }
20 }
21 void pop()
22 {
23     if(top==-1)
24         printf("stack underflow\n");
25     else
26     {
27         check(stk1[top]);
28         top--;
29     }
30 }
31 }
32 int check(int x)
33 {
34     int i,flag=0;
35     for(i=0;i<n;i++)
36     {
37         if(x==stk1[i])
38             flag++;
39     }
40     if(flag>1)
41     {
42     }
43     else{
44         stk2[k]=x;k++;
45     }
46 }
47
48 void main()
49 {
50     create();
51     int i;
52     for(i=0;i<n;i++)
```

This screenshot continues the same C program 'main.c' from the previous one. The code is identical up to line 52, where the 'main' function begins.

The screenshot shows the OnlineGDB beta interface. On the left, a sidebar menu includes 'Welcome, kulvir06', 'DSA ASSIGNMENT 1', 'Create New Project', 'My Projects', 'Learn Programming', 'Programming Questions', 'Jobs new', and 'Logout'. Below the sidebar are social media icons for Facebook, Twitter, and a plus sign, with a count of '52.9K'.

The main workspace displays a C code editor with the file 'main.c' containing:

```

33 - {int i,flag=0;
34 -     for(i=0;i<n;i++)
35 -     {
36 -         if(x==stk1[i])
37 -             flag++;
38 -     }
39 -     if(flag>1)
40 -     {
41 -     }
42 -     else{
43 -
44 -         stk2[k]=x;k++;
45 -     }
46 -
47 void main()
48 {
49     create();
50     int i;
51     for(i=0;i<n;i++)
52     push();
53     for(i=0;i<n;i++)
54     pop();
55     printf("required output:\n");
56     for(i=k-1;i>=0;i--)
57     printf("%d \t",stk2[i]);
58
59 }

```

The status bar at the bottom right shows 'ENG 2:55 PM IN 12/18/2019'.

The screenshot shows the execution results of the C program. The terminal window displays the following output:

```

enter size of stack
9
enter element in stack
2
enter element in stack
3
enter element in stack
4
enter element in stack
1
enter element in stack
1
enter element in stack
4
enter element in stack
7
enter element in stack
8
enter element in stack
2
required output:
3      7      8
...Program finished with exit code 3
Press ENTER to exit console.

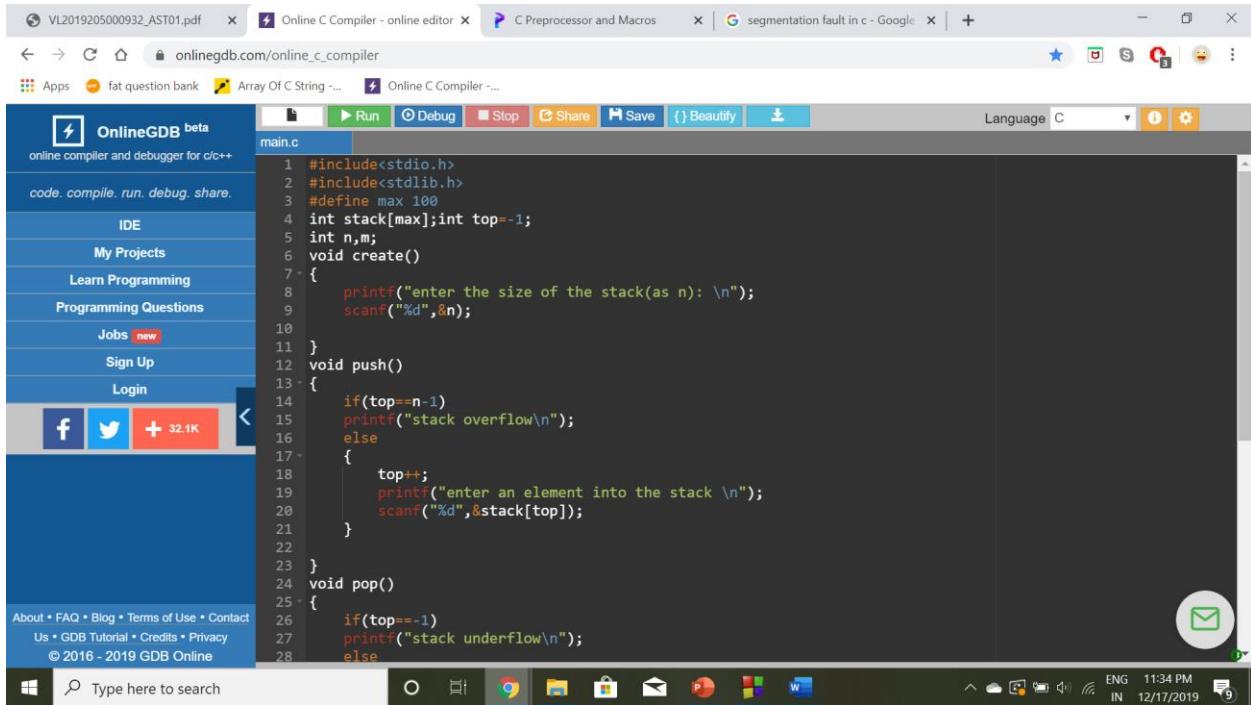
```

The status bar at the bottom right shows 'ENG 2:55 PM IN 12/18/2019'.

9>Implement a stack S of n elements using arrays. Write functions to perform PUSH and POP operations. Implement queries using push and pop functions to

- a. Retrieve the mth element of the stack S from the top ($m < n$), leaving the stack without its top $m - 1$ elements.**
- b. Retain only the elements in the odd position of the stack and pop out all even positioned elements.**

9a>



VL2019205000932_AST01.pdf Online C Compiler - online editor C Preprocessor and Macros segmentation fault in c - Google

OnlineGDB beta

IDE My Projects Learn Programming Programming Questions Jobs Sign Up Login

About • FAQ • Blog • Terms of Use • Contact Us • GDB Tutorial • Credits • Privacy © 2016 - 2019 GDB Online

main.c

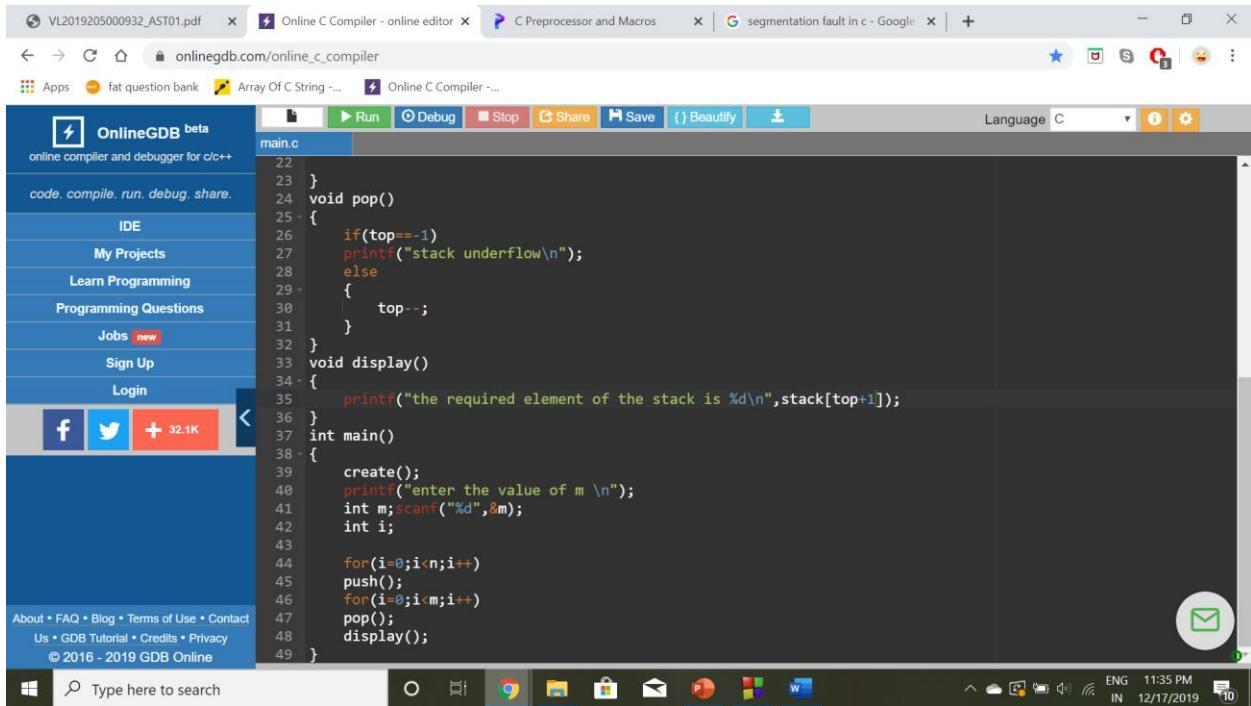
```
1 #include<stdio.h>
2 #include<stdlib.h>
3 #define max 100
4 int stack[max];int top=-1;
5 int n,m;
6 void create()
7 {
8     printf("enter the size of the stack(as n): \n");
9     scanf("%d",&n);
10 }
11 void push()
12 {
13     if(top==n-1)
14         printf("stack overflow\n");
15     else
16     {
17         top++;
18         printf("enter an element into the stack \n");
19         scanf("%d",&stack[top]);
20     }
21 }
22 void pop()
23 {
24     if(top==-1)
25         printf("stack underflow\n");
26     else
27     {
28         top--;
29     }
30 }
31 void display()
32 {
33     printf("the required element of the stack is %d\n",stack[top+1]);
34 }
35 int main()
36 {
37     create();
38     printf("enter the value of m \n");
39     int m;scanf("%d",&m);
40     int i;
41
42     for(i=0;i<n;i++)
43         push();
44     for(i=0;i<m;i++)
45         pop();
46     display();
47 }
```

Language: C

Run Debug Stop Share Save Beautify Download

Facebook Twitter + 32.1K

ENG 11:34 PM IN 12/17/2019



VL2019205000932_AST01.pdf Online C Compiler - online editor C Preprocessor and Macros segmentation fault in c - Google

OnlineGDB beta

IDE My Projects Learn Programming Programming Questions Jobs Sign Up Login

About • FAQ • Blog • Terms of Use • Contact Us • GDB Tutorial • Credits • Privacy © 2016 - 2019 GDB Online

main.c

```
22 }
23 void pop()
24 {
25     if(top==-1)
26         printf("stack underflow\n");
27     else
28     {
29         top--;
30     }
31 }
32 void display()
33 {
34     printf("the required element of the stack is %d\n",stack[top+1]);
35 }
36 int main()
37 {
38     create();
39     printf("enter the value of m \n");
40     int m;scanf("%d",&m);
41     int i;
42
43     for(i=0;i<n;i++)
44         push();
45     for(i=0;i<m;i++)
46         pop();
47     display();
48 }
```

Language: C

Run Debug Stop Share Save Beautify Download

Facebook Twitter + 32.1K

ENG 11:35 PM IN 12/17/2019

enter the size of the stack(as n):
6
enter the value of m
4
enter an element into the stack
54
enter an element into the stack
76
enter an element into the stack
82
enter an element into the stack
762
enter an element into the stack
12
the required element of the stack is 82

...Program finished with exit code 0
Press ENTER to exit console.

9b>

```
#include<stdio.h>
#include<stdlib.h>
#define max 100
int stack[max];
int n;int top=-1;
void create()
{
    printf("enter size of stack\n");
    scanf("%d",&n);
}
void push()
{
    if(top==n-1)
        printf("stack overflow");
    else
    {
        top++;
        scanf("%d",&stack[top]);
    }
}
void pop(int x)
{
    if(top==-1)
        printf("stack underflow");
    else
    {
        printf("%d ",stack[x]);
    }
}
```

The screenshot shows the OnlineGDB beta IDE interface. On the left, there's a sidebar with links like 'IDE', 'My Projects', 'Learn Programming', 'Programming Questions', 'Jobs', 'Sign Up', and 'Login'. Below the sidebar are social media sharing buttons for Facebook, Twitter, and LinkedIn, and a note that it has 32.1K users. The main area displays a C program named 'main.c'.

```

13     if(top==n-1)
14         printf("stack overflow");
15     else
16     {
17         top++;
18         scanf("%d",&stack[top]);
19     }
20 }
void pop(int x)
21 {
22     if(top==-1)
23         printf("stack underflow");
24     else
25     {
26         printf("%d ",stack[x]);
27     }
28 }
int main()
29 {
30     create();
31     int i;
32     for(i=0;i<n;i++)
33         push();
34     printf("even positioned elements\n");
35     for(i=0;i<n;i++)
36         if(i%2==1)
37             pop(i);
38 }
39 }
40 }

```

The status bar at the bottom shows the date as December 18, 2019, and the time as 12:06 AM.

This screenshot shows the execution results of the C program. A tooltip from 'C Preprocessor and Macros' points to the 'input' field where 'enter size of stack' was typed. The output window shows the program's execution:

```

input
enter size of stack
5
1
2
3
4
5
even positioned elements
2 4
...Program finished with exit code 0
Press ENTER to exit console.

```

The status bar at the bottom shows the date as December 18, 2019, and the time as 12:07 AM.

10>.Implement an ADT for a buffer in a text editor. It should support the following operations:

- **insert(c):** insert character c at cursor
- **delete():** delete and return the character at the cursor
- **left():** move the cursor one position to the left
- **right():** move the cursor one position to the right
- **get(i):** return the ith character in the buffer

GDB online Debugger | Compiler x +

onlinegdb.com

Apps fat question bank Array Of C String ... Online C Compiler ...

OnlineGDB beta

Welcome, **kulvir06**

Create New Project

My Projects

Learn Programming

Programming Questions

Jobs new

Logout

f Twitter + 52.9K

About • FAQ • Blog • Terms of Use • Contact

Us • GDB Tutorial • Credits • Privacy

© 2016 - 2019 GDB Online

main.c

```
1 #include<stdio.h>
2 #include<stdlib.h>
3 #define max 100
4 char ch[max];int top=-1,n;
5 create()
6 {
7     printf("define the maximum size for the ADT\n");
8     scanf("%d",&n);
9 }
10 void insert(char c)
11 {
12     if(top==n-1)
13         printf("maximum limit reached\n");
14     else
15     {
16         top++;
17         ch[top]=c;
18     }
19 }
20 char delete()
21 {
22     if(top==-1)
23         printf("error/:empty//cannot delete\n");
24     else
25     {
26         top--;
27         return ch[top+1];
28     }
}
```

File Explorer

Language C

3:46 PM IN 12/18/2019

GDB online Debugger | Compiler x +

onlinegdb.com

Apps fat question bank Array Of C String ... Online C Compiler ...

OnlineGDB beta

Welcome, **kulvir06**

Create New Project

My Projects

Learn Programming

Programming Questions

Jobs new

Logout

f Twitter + 52.9K

About • FAQ • Blog • Terms of Use • Contact

Us • GDB Tutorial • Credits • Privacy

© 2016 - 2019 GDB Online

main.c

```
26     top--;
27     return ch[top+1];
28 }
29 } void left()
30 {
31     if(top==0)
32         printf("error/:cannot perform operation\n");
33     else
34         top--;
35 }
36 void right()
37 {
38     if(top==n-1)
39         printf("error/:cannot perform operation\n");
40     else
41         top++;
42 }
43 char get(int i)
44 {
45     if(i>=0&&i<=n-1)
46         return ch[i];
47     else
48         return "error";
49 }
50 void main()
51 {
52     create();
53 }
```

File Explorer

Language C

3:46 PM IN 12/18/2019

GDB online Debugger | Compiler x +

onlinegdb.com

Apps fat question bank Array Of C String ... Online C Compiler ...

OnlineGDB beta

Welcome, **kulvir06**

Create New Project

My Projects

Learn Programming

Programming Questions

Jobs new

Logout

f Twitter + 52.9K

About • FAQ • Blog • Terms of Use • Contact
Us • GDB Tutorial • Credits • Privacy
© 2016 - 2019 GDB Online

main.c

```
37 void right()
38 {
39     if(top==n-1)
40         printf("error/:cannot perform operation\n");
41     else
42         top++;
43 }
44 char get(int i)
45 {
46     if(i>=0&&i<=n-1)
47         return ch[i];
48     else
49         return "error";
50 }
51 void main()
52 {
53     create();
54     delete();
55     insert('v');
56     insert('i');
57     insert('r');
58     right();
59     left();
60     insert('k');
61     delete();
62     printf("%c is in the 0th position\n",get(0));
63     printf("%c is in the 2nd position\n",get(2));
64 }
```

Language C

Run Debug Stop Save Beautify

3:47 PM IN 12/18/2019

GDB online Debugger | Compiler x +

onlinegdb.com

Apps fat question bank Array Of C String ... Online C Compiler ...

OnlineGDB beta

Welcome, **kulvir06**

Create New Project

My Projects

Learn Programming

Programming Questions

Jobs new

Logout

f Twitter + 52.9K

About • FAQ • Blog • Terms of Use • Contact
Us • GDB Tutorial • Credits • Privacy
© 2016 - 2019 GDB Online

main.c:5:1: warning: return type defaults to 'int' [-Wimplicit-int]
main.c:49:12: warning: return makes integer from pointer without a cast [-Wint-conversion]

define the maximum size for the ADT

3

error/:empty//cannot delete

error/:cannot perform operation

v is in the 0th position

k is in the 2nd position

...Program finished with exit code 25

Press ENTER to exit console.

input

Microsoft Store

3:47 PM IN 12/18/2019