**Q1 Write a program to insert element into the stack and display the element.**

**Code:**

**/\* name:Dinal Rathod**

**roll\_no:59**

**class:FY-A\*/**

#include&lt;conio.h&gt;

#include&lt;stdio.h&gt;

#define SIZE 4

void push();

void show();

int top = -1, inp\_array[SIZE];

void main()

{

int choice;

clrscr();

while (1)

{

printf(&quot;\nPerform operations on the stack:&quot;);

printf(&quot;\n 1.Push the element\n 2.Show\n 3.End&quot;);

printf(&quot;\n\nEnter the choice: &quot;);

scanf(&quot;%d&quot;, &amp;choice);

switch (choice)

{

case 1:

push();

break;

case 2:

show();

break;

case 3:

exit(0);

default:

printf(&quot;\nInvalid choice!!&quot;);

}

}

}

void push()

{

int x;

if (top == SIZE - 1)

{

printf(&quot;\nOverflow!!&quot;);

}

else

{

printf(&quot;\nEnter the element to be added on to the stack: &quot;);

scanf(&quot;%d&quot;, &amp;x);

top = top + 1;

inp\_array[top] = x;

}

}

void show()

{

int i;

if (top == -1)

{

printf(&quot;\nUnderflow!!&quot;);

}

else

{

printf(&quot;\nElements present in the stack: \n&quot;);

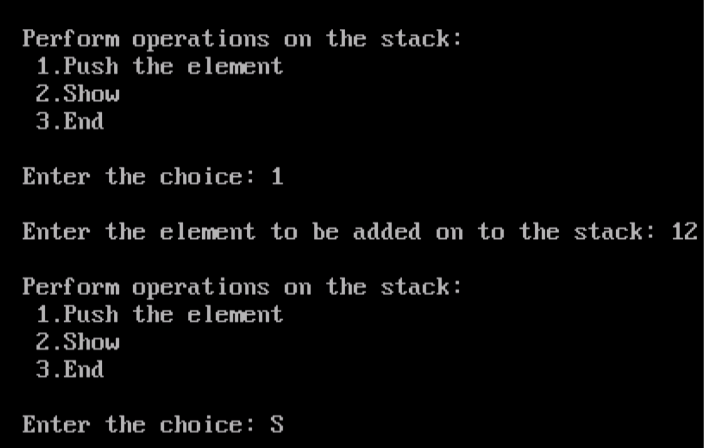
for (i = top; i &gt;= 0; --i)

printf(&quot;%d\n&quot;, inp\_array[i]);

}

}

**Output:**



**Q2. Write a program to delete an element into the stack and display the element.**

**Code:**

**/\* name:Dinal Rathod**

**roll\_no:59**

**class:FY-A\*/**

#include &lt;stdio.h&gt;

#include &lt;conio.h&gt;

#define SIZE 4

int top = -1, inp\_array[SIZE];

void push();

void pop();

void show();

int main()

{

int choice;

clrscr();

while (1)

{

printf(&quot;\nPerform operations on the stack:&quot;);

printf(&quot;\n 1.Push the element\n 2.Pop the element\n 3.Show\n

4.End&quot;);

printf(&quot;\n\nEnter the choice: &quot;);

scanf(&quot;%d&quot;, &amp;choice);

switch (choice)

{

case 1:

push();

break;

case 2:

pop();

break;

case 3:

show();

break;

case 4:

exit(0);

default:

printf(&quot;\nInvalid choice!!&quot;);

}

}

}

void push()

{

int x;

if (top == SIZE - 1)

{

printf(&quot;\nOverflow!!&quot;);

}

else

{

printf(&quot;\nEnter the element to be added onto the stack: &quot;);

scanf(&quot;%d&quot;, &amp;x);

top = top + 1;

inp\_array[top] = x;

}

}

void pop()

{

if (top == -1)

{

printf(&quot;\nUnderflow!!&quot;);

}

else

{

printf(&quot;\nPopped element: %d&quot;, inp\_array[top]);

top = top - 1;

}

}

void show()

{

int i;

if (top == -1)

{

printf(&quot;\nUnderflow!!&quot;);

}

else

{

printf(&quot;\nElements present in the stack: \n&quot;);

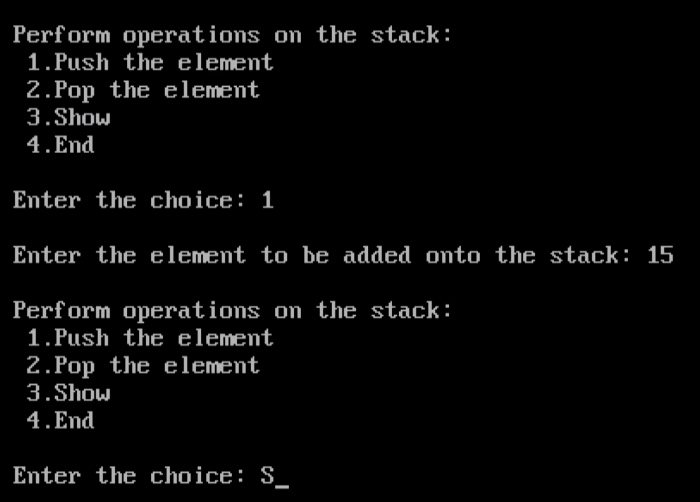
for (i = top; i &gt;= 0; --i)

printf(&quot;%d\n&quot;, inp\_array[i]);

}

}

**Output:**



**Q3. Write a program to insert element into the queue and display the element.**

**Code:**

**/\* name:Dinal Rathod**

**roll\_no:59**

**class:FY-A\*/**

#include &lt;stdio.h&gt;

# define SIZE 5

void enqueue();

void show();

int inp\_arr[SIZE];

int Rear = - 1;

int Front = - 1;

void main()

{

int ch;

clrscr();

while (1)

{

printf(&quot;1.Enqueue Operation\n&quot;);

printf(&quot;2.Display the Queue\n&quot;);

printf(&quot;3.Exit\n&quot;);

printf(&quot;\nEnter your choice of operations : &quot;);

scanf(&quot;%d&quot;, &amp;ch);

switch (ch)

{

case 1:

enqueue();

break;

case 2:

show();

break;

case 3:

exit(0);

default:

printf(&quot;Incorrect choice \n&quot;);

}

}

}

void enqueue()

{

int insert\_item;

if (Rear == SIZE - 1)

printf(&quot;Overflow \n&quot;);

else

{

if (Front == - 1)

Front = 0;

printf(&quot;Element to be inserted in the Queue\n : &quot;);

scanf(&quot;%d&quot;, &amp;insert\_item);

Rear = Rear + 1;

inp\_arr[Rear] = insert\_item;

}

}

void show()

{

int i;

if (Front == - 1)

printf(&quot;Empty Queue \n&quot;);

else

{

printf(&quot;Queue: \n&quot;);

for (i = Front; i &lt;= Rear; i++)

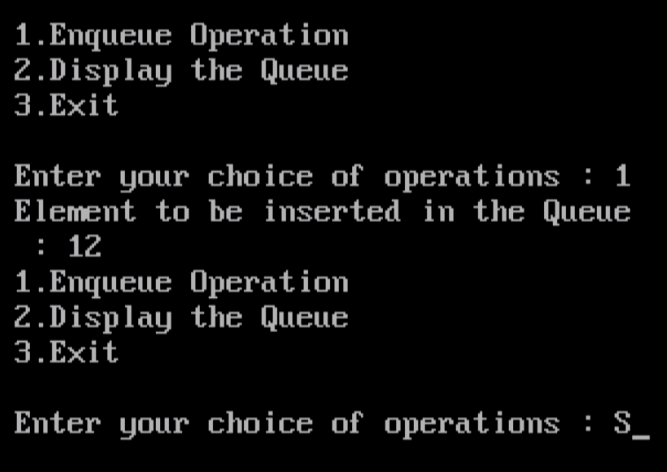
printf(&quot;%d &quot;, inp\_arr[i]);

printf(&quot;\n&quot;);

}

}

**Output:**



**Q4. Write a program to delete an element into the queue and display the element.**

**Code:**

**/\* name:Dinal Rathod**

**roll\_no:59**

**class:FY-A\*/**

#include &lt;stdio.h&gt;

# define SIZE 5

void enqueue();

void dequeue();

void show();

int inp\_arr[SIZE];

int Rear = - 1;

int Front = - 1;

void main()

{

int ch;

clrscr();

while (1)

{

printf(&quot;1.Enqueue Operation\n&quot;);

printf(&quot;2.Dequeue Operation\n&quot;);

printf(&quot;3.Display the Queue\n&quot;);

printf(&quot;4.Exit\n&quot;);

printf(&quot;Enter your choice of operations : &quot;);

scanf(&quot;%d&quot;, &amp;ch);

switch (ch)

{

case 1:

enqueue();

break;

case 2:

dequeue();

break;

case 3:

show();

break;

case 4:

exit(0);

default:

printf(&quot;Incorrect choice \n&quot;);

}

}

}

void enqueue()

{

int insert\_item;

if (Rear == SIZE - 1)

printf(&quot;Overflow \n&quot;);

else

{

if (Front == - 1)

Front = 0;

printf(&quot;Element to be inserted in the Queue\n : &quot;);

scanf(&quot;%d&quot;, &amp;insert\_item);

Rear = Rear + 1;

inp\_arr[Rear] = insert\_item;

}

}

void dequeue()

{

if (Front == - 1 || Front &gt; Rear)

{

printf(&quot;Underflow \n&quot;);

return ;

}

else

{

printf(&quot;Element deleted from the Queue: %d\n&quot;, inp\_arr[Front]);

Front = Front + 1;

}

}

void show()

{

int i;

if (Front == - 1)

printf(&quot;Empty Queue \n&quot;);

else

{

printf(&quot;Queue: \n&quot;);

for (i = Front; i &lt;= Rear; i++)

printf(&quot;%d &quot;, inp\_arr[i]);

printf(&quot;\n&quot;);

}

}

**Output:**

