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

**Code:**

/\* name:gohel ruchi

roll\_no:17

class:FY-A\*/

#include<conio.h>

#include<stdio.h>

#define SIZE 4

void push();

void show();

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

void main()

{

int choice;

clrscr();

printf("\nname:gohel ruchi\nroll\_no.:17\nclass:FY-A\n\n");

while (1)

{

printf("\nPerform operations on the stack:");

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

printf("\n\nEnter the choice: ");

scanf("%d", &choice);

switch (choice)

{

case 1:

push();

break;

case 2:

show();

break;

case 3:

exit(0);

default:

printf("\nInvalid choice!!");

}

}

}

void push()

{

int x;

if (top == SIZE - 1)

{

printf("\nOverflow!!");

}

else

{

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

scanf("%d", &x);

top = top + 1;

inp\_array[top] = x;

}

}

void show()

{

int i;

if (top == -1)

{

printf("\nUnderflow!!");

}

else

{

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

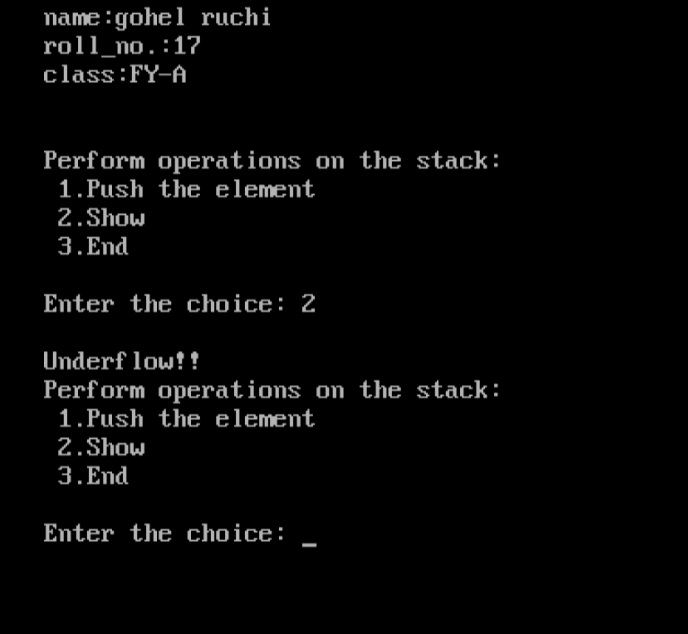
for (i = top; i >= 0; --i)

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

}

}

**Output:**

****

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

**Code:**

/\* name:gohel ruchi

roll\_no:17

class:FY-A\*/

#include <stdio.h>

#include <conio.h>

#define SIZE 4

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

void push();

void pop();

void show();

int main()

{

int choice;

clrscr();

printf("\nname:gohel ruchi\nroll\_no.:17s\nclass:FY-A\n\n");

while (1)

{

printf("\nPerform operations on the stack:");

printf("\n 1.Push the element\n 2.Pop the element\n 3.Show\n 4.End");

printf("\n\nEnter the choice: ");

scanf("%d", &choice);

switch (choice)

{

case 1:

push();

break;

case 2:

pop();

break;

case 3:

show();

break;

case 4:

exit(0);

default:

printf("\nInvalid choice!!");

}

}

}

void push()

{

int x;

if (top == SIZE - 1)

{

printf("\nOverflow!!");

}

else

{

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

scanf("%d", &x);

top = top + 1;

inp\_array[top] = x;

}

}

void pop()

{

if (top == -1)

{

printf("\nUnderflow!!");

}

else

{

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

top = top - 1;

}

}

void show()

{

int i;

if (top == -1)

{

printf("\nUnderflow!!");

}

else

{

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

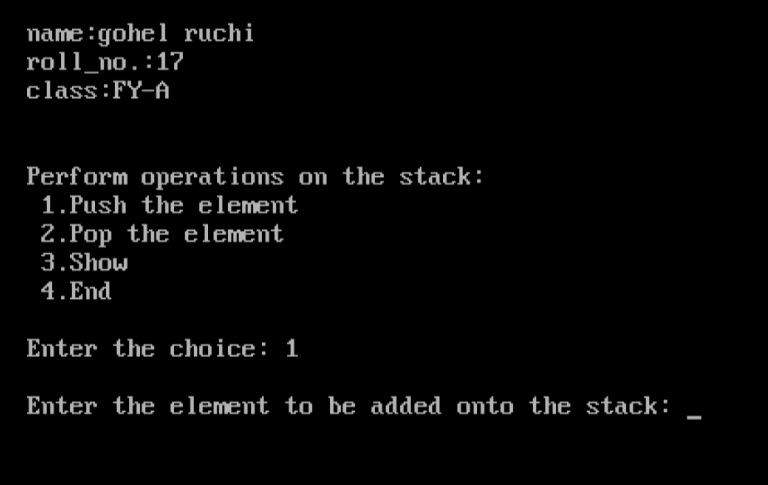
for (i = top; i >= 0; --i)

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

}

}

**Output:**



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

**Code:**

/\* name:gohel ruchi

roll\_no:17

class:FY-A\*/

#include <stdio.h>

# define SIZE 5

void enqueue();

void show();

int inp\_arr[SIZE];

int Rear = - 1;

int Front = - 1;

void main()

{

int ch;

clrscr();

printf("\nname:gohel ruchi\nroll\_no.:17\nclass:FY-A\n\n\n");

while (1)

{

printf("1.Enqueue Operation\n");

printf("2.Display the Queue\n");

printf("3.Exit\n");

printf("\nEnter your choice of operations : ");

scanf("%d", &ch);

switch (ch)

{

case 1:

enqueue();

break;

case 2:

show();

break;

case 3:

exit(0);

default:

printf("Incorrect choice \n");

}

}

}

void enqueue()

{

int insert\_item;

if (Rear == SIZE - 1)

printf("Overflow \n");

else

{

if (Front == - 1)

Front = 0;

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

scanf("%d", &insert\_item);

Rear = Rear + 1;

inp\_arr[Rear] = insert\_item;

}

}

void show()

{

int i;

if (Front == - 1)

printf("Empty Queue \n");

else

{

printf("Queue: \n");

for (i = Front; i <= Rear; i++)

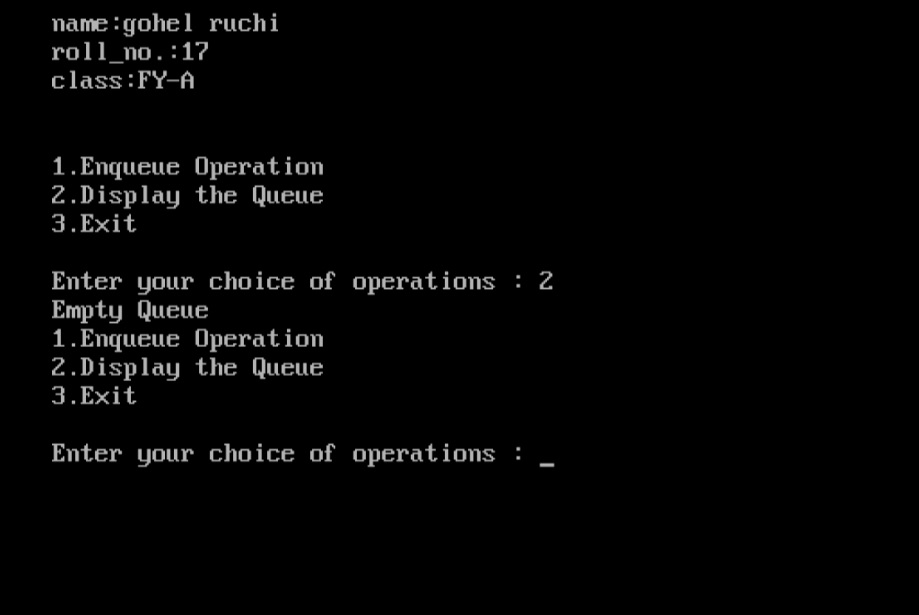
printf("%d ", inp\_arr[i]);

printf("\n");

}

}

**Output:**

****

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

**Code:**

/\* name: gohel ruchi

roll\_no:17

class:FY-A\*/

#include <stdio.h>

# define SIZE 5

void enqueue();

void dequeue();

void show();

int inp\_arr[SIZE];

int Rear = - 1;

int Front = - 1;

void main()

{

int ch;

clrscr();

printf("\nname:gohel ruchi\nroll\_no.:17\nclass:FY-A\n\n\n");

while (1)

{

printf("1.Enqueue Operation\n");

printf("2.Dequeue Operation\n");

printf("3.Display the Queue\n");

printf("4.Exit\n");

printf("Enter your choice of operations : ");

scanf("%d", &ch);

switch (ch)

{

case 1:

enqueue();

break;

case 2:

dequeue();

break;

case 3:

show();

break;

case 4:

exit(0);

default:

printf("Incorrect choice \n");

}

}

}

void enqueue()

{

int insert\_item;

if (Rear == SIZE - 1)

printf("Overflow \n");

else

{

if (Front == - 1)

Front = 0;

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

scanf("%d", &insert\_item);

Rear = Rear + 1;

inp\_arr[Rear] = insert\_item;

}

}

void dequeue()

{

if (Front == - 1 || Front > Rear)

{

printf("Underflow \n");

return ;

}

else

{

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

Front = Front + 1;

}

}

void show()

{

int i;

if (Front == - 1)

printf("Empty Queue \n");

else

{

printf("Queue: \n");

for (i = Front; i <= Rear; i++)

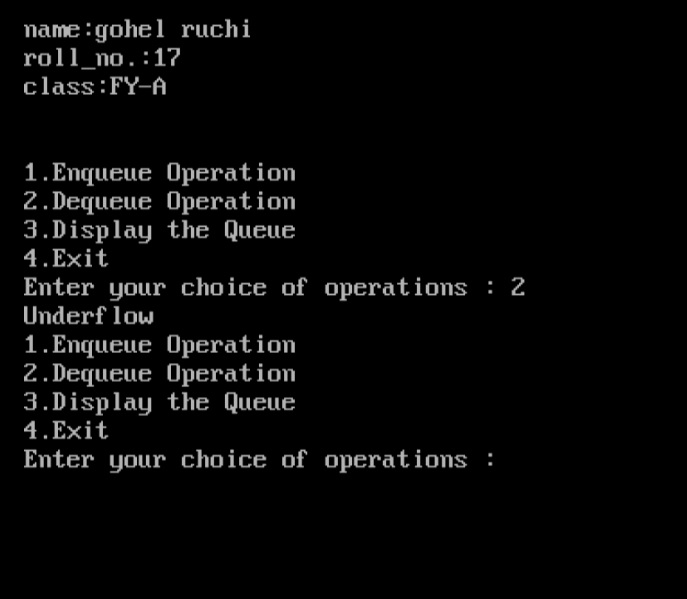
printf("%d ", inp\_arr[i]);

printf("\n");

}

}

**Output:**

****