**Program 1)**

**Write a Program to Use Stack operations to do the following:**

1. **Assign to a variable name Y the value of the third element from the top of the stack and keep the stack undisturbed.**
2. **Given an arbitrary integer n pop out the top n elements. A message should be displayed if an unusual condition is encountered.**
3. **Assign to a variable name Y the value of the third element from the bottom of the stack and keep the stack undisturbed.**

**(Hint: you may use a temporary stack)**

**Program :**

#include <stdio.h>

#define SIZE 25

struct stack

{

int top;

int data[SIZE];

};

typedef struct stack st;

void push(st \*s, int ele)

{

if (s->top == SIZE - 1)

printf("The stack is full!\n");

else

{

s->top++;

s->data[s->top] = ele;

}

}

int pop(st \*s)

{

if (s->top != -1)

{

return s->data[(s->top)--];

}

else

{

printf("Stack is empty!\n");

return -1;

}

}

int peek(st \*s)

{

if (s->top == -1)

{

printf("The stack is empty\n");

return -1;

}

else

return s->data[s->top];

}

void display(st \*s)

{

int temp;

temp = s->top;

if (s->top == -1)

printf("The stack is empty\n");

else

while (temp >= 0)

{

printf("%d\n", s->data[temp]);

temp--;

}

}

int thirdfromtop(st s)

{

int i, j;

for (i = 0; i < 2; i++)

j = pop(&s);

return peek(&s);

}

int thirdfrombottom(st s, int len)

{

int i, n;

for (i = 0; i < (len - 3); i++)

pop(&s);

return peek(&s);

}

int main()

{

st s;

int i, j, n, temp, y = 0;

s.top = -1;

printf("Enter the number of operations\n");

scanf("%d", &n);

printf("1. push\n2. pop\n3. peek\n4. display\n5. 3rd element from top\n6. 3rd element from bottom\n7. pop n elements\n");

for (i = 0; i < n; i++)

{

scanf("%d", &j);

switch (j)

{

case 1:

scanf("%d", &temp);

push(&s, temp);

break;

case 2:

temp = pop(&s);

if (temp != -1)

printf("the popped element is %d\n", temp);

break;

case 3:

temp = peek(&s);

if (temp != -1)

printf("the element at top is %d\n", temp);

break;

case 4:

display(&s);

break;

case 5:

if (s.top >= 2)

{

y = thirdfromtop(s);

printf("the 3rd element from top is : %d\n", y);

}

else

{

printf("the 3rd element from top doesnt exists\n");

}

break;

case 6:

if (s.top >= 2)

{

y = thirdfrombottom(s, s.top+1);

printf("the 3rd element from bottom is : %d\n", y);

}

else

{

printf("the 3rd element from bottom doesnt exists\n");

}

break;

case 7:

scanf("%d", &temp);

if (temp <= s.top +1)

{

for (int i = 0; i < temp; i++)

{

pop(&s);

}

}

else

printf("the number of elements in the stack is less than %d\n", temp);

break;

}

}

}

**Outputs :**

OUTPUT 1)

Demonstration of Working of data structure STACK

1:PUSH

2:POP

3:Display

4:Third Element from Top

5:Pop n Elements

6:Third Element from bottom

7:exit

Enter the Choice :1

Enter the element to be pushed 100

1:PUSH

2:POP

3:Display

4:Third Element from Top

5:Pop n Elements

6:Third Element from bottom

7:exit

Enter the Choice :1

Enter the element to be pushed 200

1:PUSH

2:POP

3:Display

4:Third Element from Top

5:Pop n Elements

6:Third Element from bottom

7:exit

Enter the Choice :1

Enter the element to be pushed 300

1:PUSH

2:POP

3:Display

4:Third Element from Top

5:Pop n Elements

6:Third Element from bottom

7:exit

Enter the Choice :3

Stack Elements are

100

200

300

1:PUSH

2:POP

3:Display

4:Third Element from Top

5:Pop n Elements

6:Third Element from bottom

7:exit

Enter the Choice :1

Enter the element to be pushed 400

1:PUSH

2:POP

3:Display

4:Third Element from Top

5:Pop n Elements

6:Third Element from bottom

7:exit

Enter the Choice :3

Stack Elements are

100

200

300

400

1:PUSH

2:POP

3:Display

4:Third Element from Top

5:Pop n Elements

6:Third Element from bottom

7:exit

Enter the Choice :4

Third Element From Top 200

1:PUSH

2:POP

3:Display

4:Third Element from Top

5:Pop n Elements

6:Third Element from bottom

7:exit

Enter the Choice :3

Stack Elements are

100

200

300

400

1:PUSH

2:POP

3:Display

4:Third Element from Top

5:Pop n Elements

6:Third Element from bottom

7:exit

Enter the Choice :2

Poped element is 400

1:PUSH

2:POP

3:Display

4:Third Element from Top

5:Pop n Elements

6:Third Element from bottom

7:exit

Enter the Choice :2

Poped element is 300

1:PUSH

2:POP

3:Display

4:Third Element from Top

5:Pop n Elements

6:Third Element from bottom

7:exit

Enter the Choice :3

Stack Elements are

100

200

1:PUSH

2:POP

3:Display

4:Third Element from Top

5:Pop n Elements

6:Third Element from bottom

7:exit

Enter the Choice :7

OUTPUT 2)

Demonstration of Working of data structure STACK

1:PUSH

2:POP

3:Display

4:Third Element from Top

5:Pop n Elements

6:Third Element from bottom

7:exit

Enter the Choice :1

Enter the element to be pushed 1

1:PUSH

2:POP

3:Display

4:Third Element from Top

5:Pop n Elements

6:Third Element from bottom

7:exit

Enter the Choice :1

Enter the element to be pushed 2

1:PUSH

2:POP

3:Display

4:Third Element from Top

5:Pop n Elements

6:Third Element from bottom

7:exit

Enter the Choice :2

Poped element is 2

1:PUSH

2:POP

3:Display

4:Third Element from Top

5:Pop n Elements

6:Third Element from bottom

7:exit

Enter the Choice :2

Poped element is 1

1:PUSH

2:POP

3:Display

4:Third Element from Top

5:Pop n Elements

6:Third Element from bottom

7:exit

Enter the Choice :2

Stack Underflow!

OUTPUT 3)

Demonstration of Working of data structure STACK

1:PUSH

2:POP

3:Display

4:Third Element from Top

5:Pop n Elements

6:Third Element from bottom

7:exit

Enter the Choice :1

Enter the element to be pushed 10

1:PUSH

2:POP

3:Display

4:Third Element from Top

5:Pop n Elements

6:Third Element from bottom

7:exit

Enter the Choice :1

Enter the element to be pushed 20

1:PUSH

2:POP

3:Display

4:Third Element from Top

5:Pop n Elements

6:Third Element from bottom

7:exit

Enter the Choice :1

Enter the element to be pushed 30

1:PUSH

2:POP

3:Display

4:Third Element from Top

5:Pop n Elements

6:Third Element from bottom

7:exit

Enter the Choice :1

Enter the element to be pushed 40

1:PUSH

2:POP

3:Display

4:Third Element from Top

5:Pop n Elements

6:Third Element from bottom

7:exit

Enter the Choice :1

Enter the element to be pushed 50

1:PUSH

2:POP

3:Display

4:Third Element from Top

5:Pop n Elements

6:Third Element from bottom

7:exit

Enter the Choice :1

Enter the element to be pushed 60

Stack Overflow