

```
1. 1.Program to demonstrate stack using arrays using c
2. #include<stdio.h>
3. #include<stdlib.h>
4. #define size 5
5. int top=-1,item;
6. int stack[size];
7. void push()
8. {
9. if(top<size-1)
10. {
11. printf("Enter the Element: ");
12. scanf("%d",&item);
13. top++;
14. stack[top]=item;
15. printf("%d is Pushed into stack\n",item);
16. }
17. else
18. {
19. printf("The stack is Overflowed\n");
20. }
21. }
22. void pop()
23. {
24. if(top== -1)
25. {
26. printf("The stack is Underflowed\n");
27. }
28. else
29. {
30. printf("%d is popped from stack\n",stack[top]);
31. top--;
32. }
33. }
34. void display()
35. {
36. if(top== -1)
37. {
38. printf("The Stack is Empty\n");
39. }
40. else
41. {
42. for(int i=0;i<=top;i++)
43. {
44. printf("%d ",stack[i]);
45. }
46. printf("\n");
47. }
48. }
49. void isfull()
50. {
51. if(top== -1)
52. {
53. printf("Stack is Empty\n");
54. }
55. else if(top==(size-1))
56. {
57. printf("Stack is Full\n");
58. }
59. else
60. {
61. printf("Stack is Not Full\n");
62. }
```

```

63. }
64. int main()
65. {
66. while(1)
67. {
68. printf(" Menu:\n 1.Push\n 2.Pop\n 3.Display\n 4.Isfull\n 5.Exit\n");
69. int n;
70. printf("Enter your Choice : ");
71. scanf("%d",&n);
72. switch(n)
73. {
74. case 1: push();
75. break;
76. case 2: pop();
77. break;
78. case 3: display();
79. break;
80. case 4: isfull();
81. break;
82. case 5: exit(1);
83. break;
84. }
85. }
86. }

```

87. output:

```

88. yasar@yasar-Lenovo-G50-80:~/Desktop/DataStructures/lords/lab$ gcc stackArrays1.cc -o
sa1

```

```

89. yasar@yasar-Lenovo-G50-80:~/Desktop/DataStructures/lords/lab$ ./sa1

```

```

90. Menu:
91. 1.Push
92. 2.Pop
93. 3.Display
94. 4.Isfull
95. 5.Exit
96. Enter your Choice : 1
97. Enter the Element: 10
98. 10 is Pushed into stack
99. Menu:
100. 1.Push
101. 2.Pop
102. 3.Display
103. 4.Isfull
104. 5.Exit
105. Enter your Choice : 1
106. Enter the Element: 20
107. 20 is Pushed into stack
108. Menu:
109. 1.Push
110. 2.Pop
111. 3.Display
112. 4.Isfull
113. 5.Exit
114. Enter your Choice : 3
115. 10 20
116. Menu:

```

117. 1.Push  
118. 2.Pop  
119. 3.Display  
120. 4.Isfull  
121. 5.Exit  
122. Enter your Choice : 1  
123. Enter the Element: 30  
124. 30 is Pushed into stack  
125. Menu:  
126. 1.Push  
127. 2.Pop  
128. 3.Display  
129. 4.Isfull  
130. 5.Exit  
131. Enter your Choice : 1  
132. Enter the Element: 40  
133. 40 is Pushed into stack  
134. Menu:  
135. 1.Push  
136. 2.Pop  
137. 3.Display  
138. 4.Isfull  
139. 5.Exit  
140. Enter your Choice : 3  
141. 10 20 30 40  
142. Menu:  
143. 1.Push  
144. 2.Pop  
145. 3.Display  
146. 4.Isfull  
147. 5.Exit  
148. Enter your Choice : 4  
149. Stack is Not Full  
150. Menu:  
151. 1.Push  
152. 2.Pop  
153. 3.Display  
154. 4.Isfull  
155. 5.Exit  
156. Enter your Choice : 1  
157. Enter the Element: 50  
158. 50 is Pushed into stack  
159. Menu:  
160. 1.Push  
161. 2.Pop  
162. 3.Display  
163. 4.Isfull  
164. 5.Exit  
165. Enter your Choice : 3  
166. 10 20 30 40 50  
167. Menu:  
168. 1.Push

```
169.      2.Pop
170.      3.Display
171.      4.Isfull
172.      5.Exit
173.      Enter your Choice : 4
174.      Stack is Full
175.      Menu:
176.      1.Push
177.      2.Pop
178.      3.Display
179.      4.Isfull
180.      5.Exit
181.      Enter your Choice : 5
182.      yasar@yasar-Lenovo-G50-80:~/Desktop/DataStructures/lords/lab$
183.
184.
185.
186.
187.
188.
189.
190.
191.
192.
193.
194.
195.
196.
197.
198.
199.
200.
201.
202.
203.
204.
```