

Stack Implementation

Jumel

Pg. 1.

```

#include <stdio.h>
int stack[100], choice, n, top, x, i;
void push(void);
void pop(void);
void display(void);
int main()
{
    top = -1;
    printf("\n Enter the size of STACK [MAX = 100] : ");
    scanf("%d", &n);
    printf("\n It STACK OPERATIONS USING ARRAY ");
    printf("\n It 1. PUSH & n 2. POP \n It 3. DISPLAY \n It 4. EXIT");
    do
    {
        printf("\n Enter the choice : ");
        scanf("%d", &choice);
        switch (choice)
        {
            case 1:
            {
                push();
                break;
            }
            case 2:
            {
                pop();
                break;
            }
            case 3:
            {
                display;
                break;
            }
        }
    }
}

```

case 4:

```
{
    printf (" \n It EXIT POINT ");
    break;
}
```

default;

```
{
    printf (" \n It Please Enter a Valid choice (1/2/3/4) ");
}
```

}

}

while (choice != 4);

return 0;

}

void (push)

{

if (top >= n-1)

{

printf (" \n It STACK is overflow ");

}

else

{

printf ("Enter a value to be pushed : ");

scanf ("%d", &x);

top ++;

stack [top] = x;

}

}

void pop()

{

```
if (top <= -1)
```

```
{
```

```
    printf("In Stack is under flow");
```

```
}
```

```
else
```

```
{
```

```
    printf("In The popped elements is %d", stack[top]);
```

```
    top--;
```

```
}
```

```
}
```

```
void display()
```

```
{
```

```
    if (top >= 0)
```

```
{
```

```
    printf("In The elements in STACK\n");
```

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

```
        printf("In %d", stack[i]);
```

```
    printf("In Present choice ");
```

```
}
```

```
else
```

```
{
```

```
    printf("In The STACK is empty");
```

```
}
```

```
}
```