

Data structure using c

Implementation of stacks using Array and Link list

Uses of stacks.

Implementation of stack using Array

```
# define N 5
int stack[N];
int top = -1;
void Push()
{
    printf("Enter data");
    scanf("%d", &x);
    if (top == N-1)
    {
        printf("overflow");
    }
    else
    {
        top++;
        stack[top] = x;
    }
}
```

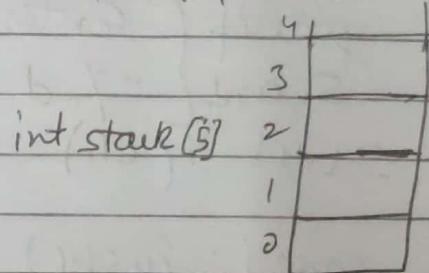
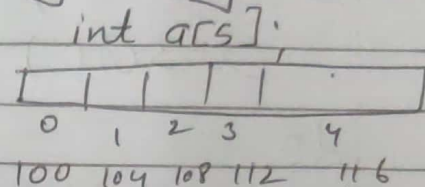
```
void Pop()
{
    int item;
    if (top == -1)
    {
        printf("underflow");
    }
    else
    {
        item = stack[top];
        top--;
        printf("%d", item);
    }
}
```

Pop()

Peek()

```
void peek()
{
    if (top == -1)
    {
        printf("stack is empty");
    }
    else
    {
        printf("%d", stack[top]);
    }
}

void display()
{
    int i;
    for (i = top; i >= 0; i--)
    {
        printf("%d", stack[i]);
    }
}
```



void main()

```
{
    Push()
    Pop()
    Peek()
    display()
}
```

```
# define N, 5
```

```
int stack[N];
```

```
int top = -1;
```

```
void main()
```

```
{ int ch;
```

```
  getch();
```

```
do
```

```
{ printf("Enter choice: 1: Push 2: Pop 3: Peek 4: Display");
```

```
  scanf("%d", &ch);
```

```
  switch(ch)
```

```
{
```

```
  case 1: Push();
```

```
    break;
```

```
  case 2: Pop();
```

```
    break;
```

```
  case 3: Peek();
```

```
    break;
```

```
  case 4: display();
```

```
    break;
```

```
  default: printf("Invalid choice");
```

```
}
```

```
while (ch != 0);
```

```
  getch();
```

```
}
```

Time complexity

order of 1

$O(1)$