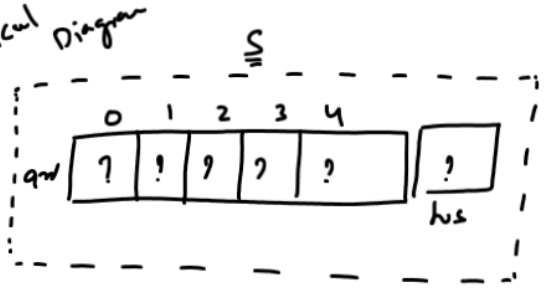
First version

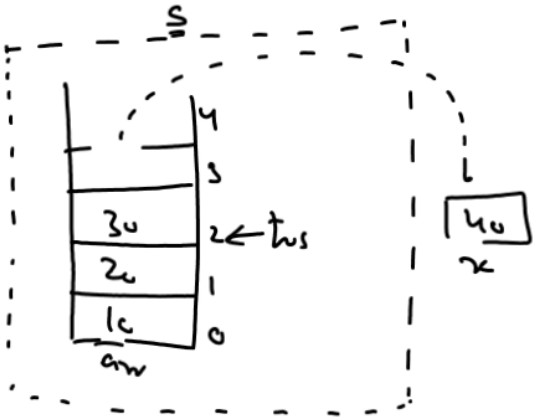
Implementing Stack In C

```
struct Stack
{
    int arr[5];
    int tos;
};
int main( )
{ int x;
  struct Stack S;
   tos=-1; X
   S.tos=-1;
```

physical Diagram



Logical Diagram




```
S.tos = S.tos+1;
S.arr[S.tos ] =10;

S.tos = S.tos + 1;
S.arr[S.tos]= 20;
  .
  .
  .
S.tos = S.tos+1;
S.arr[S.tos]= 50;
```

push

```
x = S.arr[S.tos];
S.tos= S.tos - 1;
printf("\n Popped ele = %d", x);

x = S.arr[S.tos];
S.tos = S.tos - 1;
printf("\n Popped ele = %d", x);
  .
  .
  .
return 0;
}
```

Pop

Second Version

```
struct Stack
{   int arr[5];
    int tos;
};
int main()
{
  struct Stack S;
  int i, x;
  S.tos=-1;
  for(i=1; i<=5; i++)
    { printf("Enter no to push:");
      scanf("%d",&x);
      S.tos = S.tos+1;
      S.arr[S.tos]=x;
    }
  for(i=1; i<=5; i++)
    {
      x = S.arr[S.tos];
      S.tos = S.tos - 1;
      printf("\n Popped ele=%d",x);
    }
  return 0;
}
```

Third version

```
struct Stack
{   int arr[5];
    int tos;
};
void push(struct Stack *, int);
int pop(struct Stack *);

int main()
{
    struct Stack S;
    int i, x;
    S.tos=-1;
```

] prototypes / decl of functions

```
void disply(int *);
int main()
{
    int a=10;
    display(&a);
    fun(" %d ", a);
}
void display(int *p)
{
    fun(" %d", *p);
    *p=*p+1;
}
```