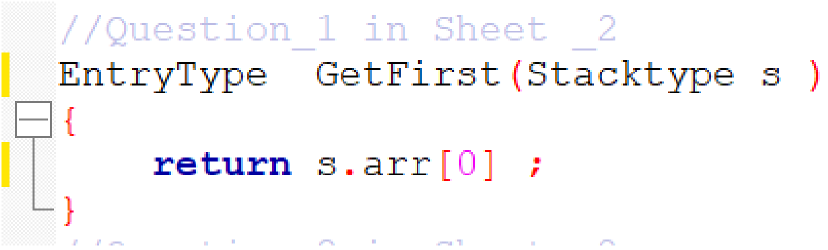
Ans_Sheet_2

1) Write a function that returns the first element entered to a stack. (implementation level)

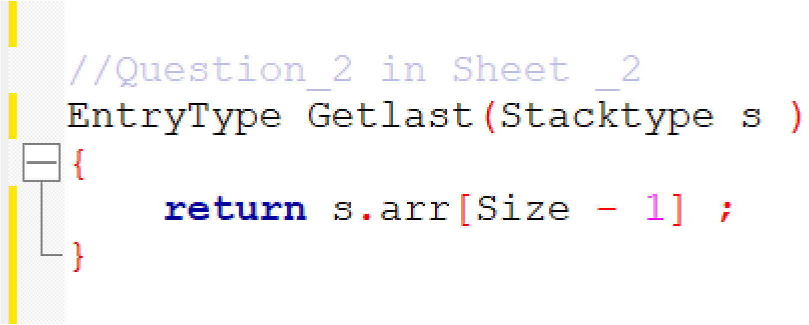
Ans



```
//Question_1 in Sheet _2
EntryType  GetFirst(Stacktype s )
{
    return s.arr[0] ;
}
```

2) Write a function that returns a copy from the last element in a stack. (implementation level)

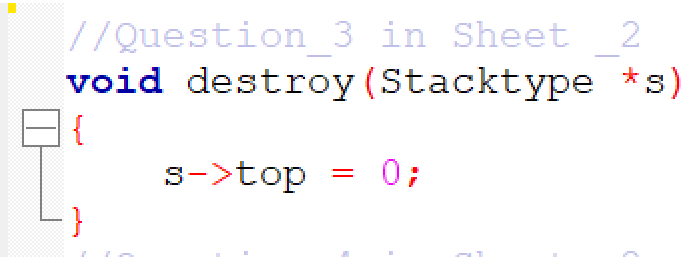
Ans



```
//Question_2 in Sheet _2
EntryType Getlast(Stacktype s )
{
    return s.arr[Size - 1] ;
}
```

3) Write a function to destroy a stack. (implementation level)

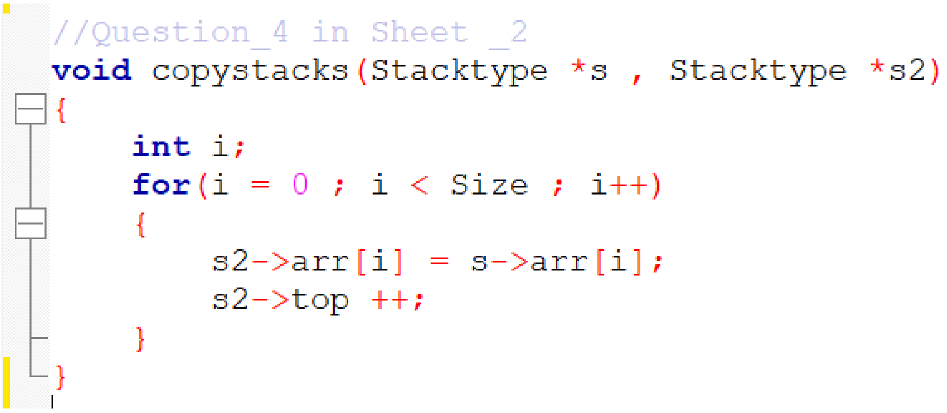
Ans



```
//Question_3 in Sheet _2
void destroy(Stacktype *s)
{
    s->top = 0;
}
```

4) Write a function to copy a stack to another. (implementation level)

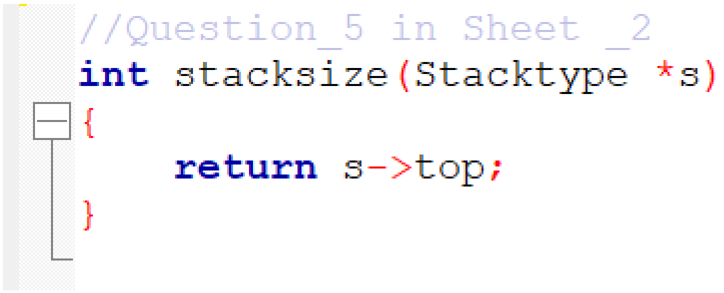
Ans



```
//Question_4 in Sheet _2
void copystacks(Stacktype *s , Stacktype *s2)
{
    int i;
    for(i = 0 ; i < Size ; i++)
    {
        s2->arr[i] = s->arr[i];
        s2->top ++;
    }
}
```

5) Write a function to return the size of a stack (implementation level)

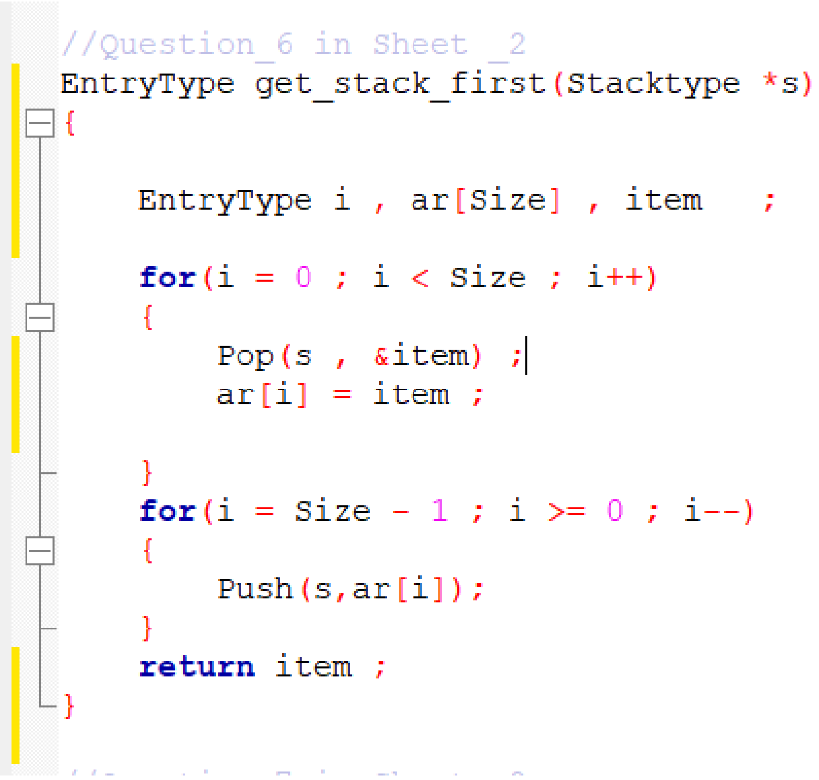
Ans



```
//Question_5 in Sheet _2
int stacksize(Stacktype *s)
{
    return s->top;
}
```

6) Write a function that returns the first element entered to a stack. (user level)

Ans

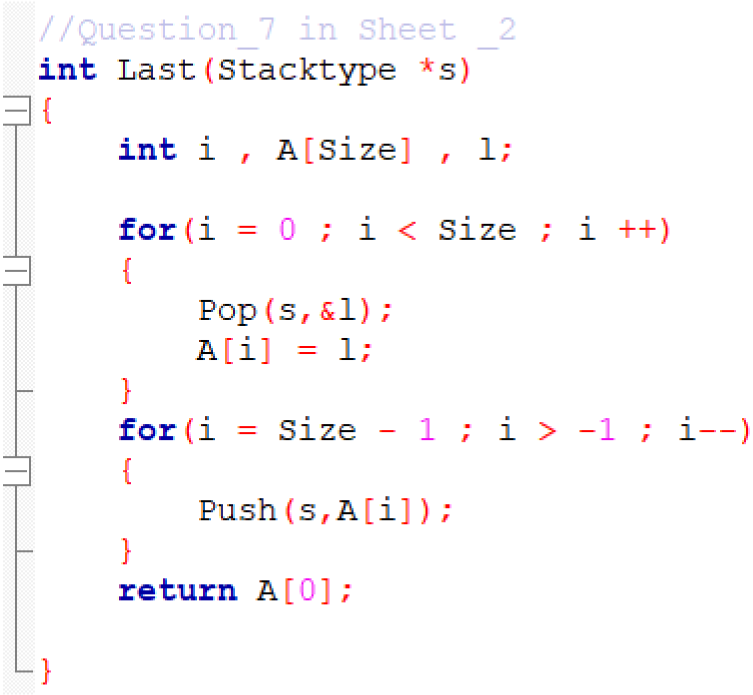


```
//Question_6 in Sheet _2
EntryType get_stack_first(Stacktype *s)
{

    EntryType i , ar[Size] , item   ;

    for(i = 0 ; i < Size ; i++)
    {
        Pop(s , &item) ;
        ar[i] = item ;

    }
    for(i = Size - 1 ; i >= 0 ; i--)
    {
        Push(s,ar[i]);
    }
    return item ;
}
```

---

7) Write a function that returns a copy from the last element in a stack. (user level)

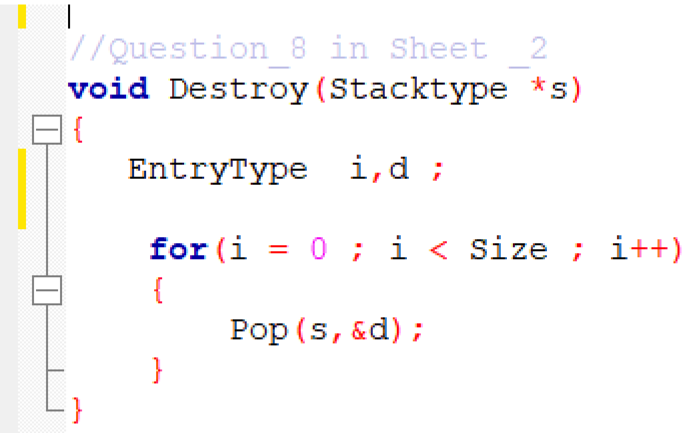
Ans



```
//Question_7 in Sheet _2
int Last(Stacktype *s)
{
    int i , A[Size] , l;

    for(i = 0 ; i < Size ; i ++)
    {
        Pop(s,&l);
        A[i] = l;
    }
    for(i = Size - 1 ; i > -1 ; i--)
    {
        Push(s,A[i]);
    }
    return A[0];

}
```

8) Write a function to destroy a stack. (user level)

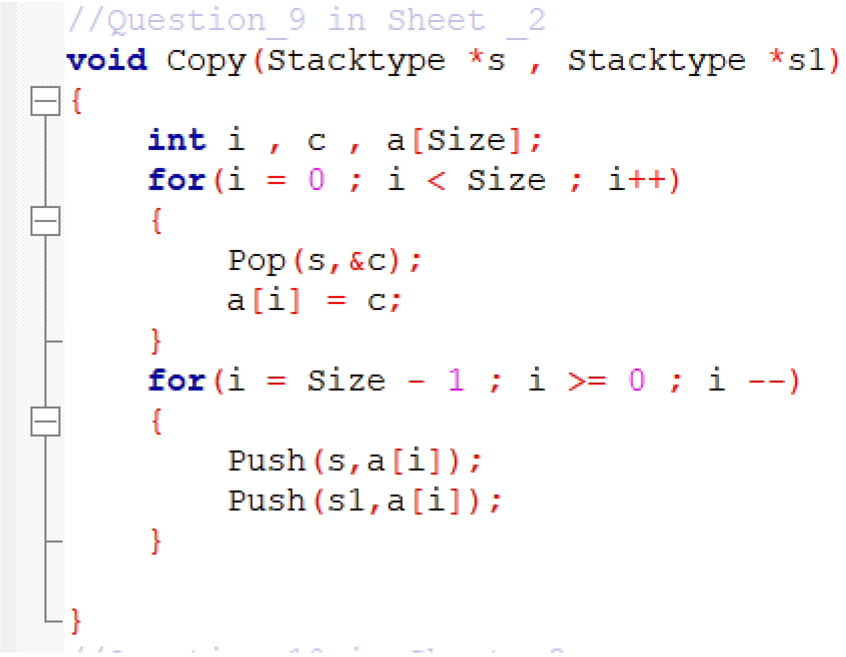
Ans



```
//Question_8 in Sheet _2
void Destroy(Stacktype *s)
{
   EntryType  i,d ;

    for(i = 0 ; i < Size ; i++)
    {
        Pop(s,&d);
    }
}
```

9) Write a function to copy a stack to another. (user level)

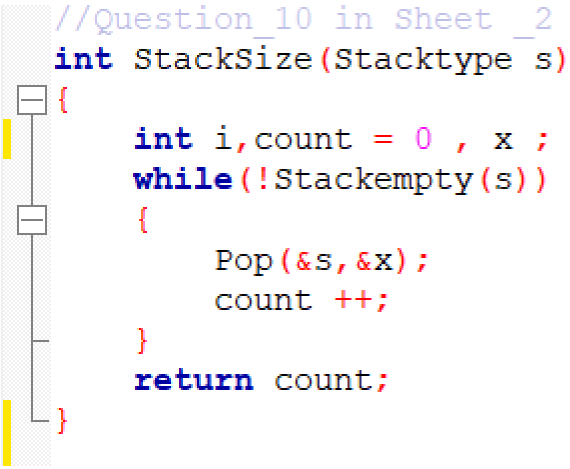
Ans



```
//Question_9 in Sheet _2
void Copy(Stacktype *s , Stacktype *s1)
{
    int i , c , a[Size];
    for(i = 0 ; i < Size ; i++)
    {
        Pop(s,&c);
        a[i] = c;
    }
    for(i = Size - 1 ; i >= 0 ; i --)
    {
        Push(s,a[i]);
        Push(s1,a[i]);
    }

}
```

10) Write a function to return the size of a stack (user level)

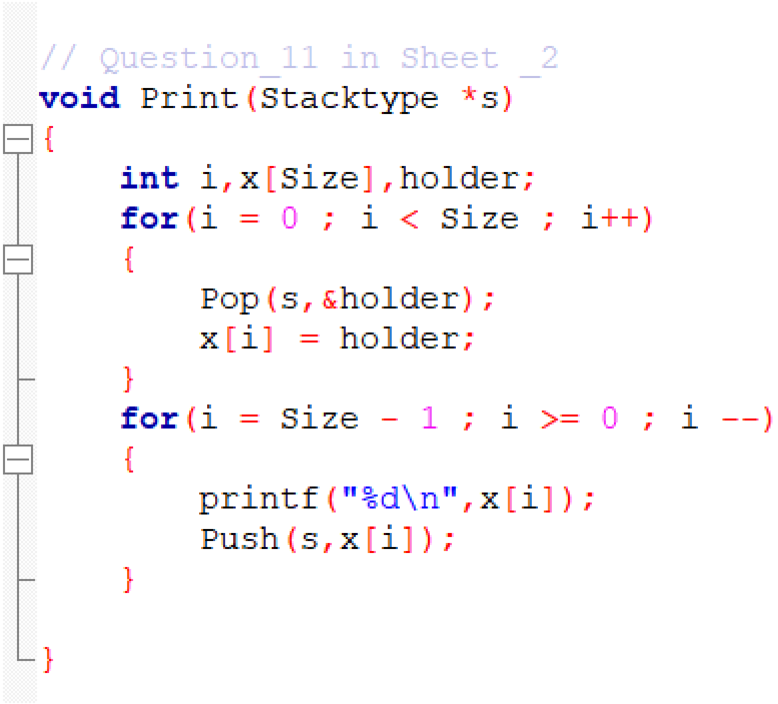
Ans



```
//Question_10 in Sheet _2
int StackSize(Stacktype s)
{
    int i,count = 0 , x ;
    while(!Stackempty(s))
    {
        Pop(&s,&x);
        count ++;
    }
    return count;
}
```

11) Write a function to print on the screen the contents of a stack without changing the stack (user level).

Ans



```
// Question_11 in Sheet _2
void Print(Stacktype *s)
{
    int i,x[Size],holder;
    for(i = 0 ; i < Size ; i++)
    {
        Pop(s,&holder);
        x[i] = holder;
    }
    for(i = Size - 1 ; i >= 0 ; i --)
    {
        printf("%d\n",x[i]);
        Push(s,x[i]);
    }

}
```