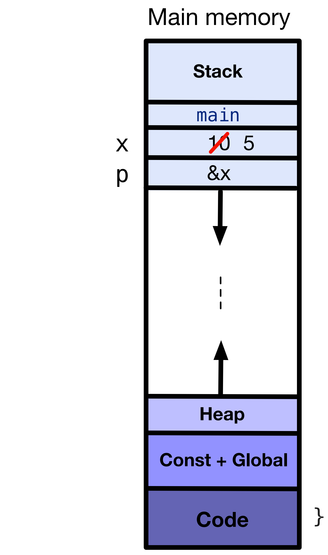
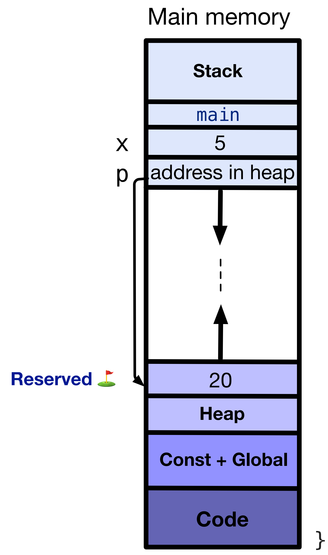



```
#include <iostream>
using namespace std;

int main(){
  int x = 10;
  int* p = NULL;
  p = &x;
  *p = 5;
```

Will reveal in the next figure

```
}
```




```
#include <iostream>
using namespace std;

int main(){
  int x = 10;
  int* p = NULL;
  p = &x;
  *p = 5;
```

1 Allocate memory for an `int` on the heap

```
  p = new int;
```

2 return its address and store it in p

```
  *p = 20;
```

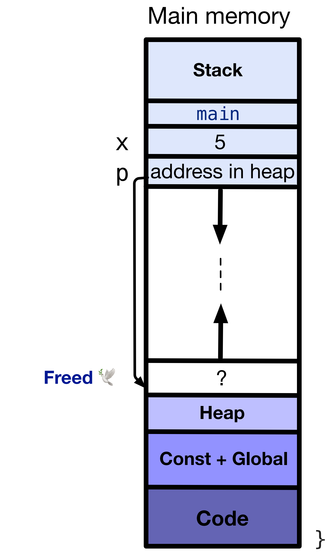
prints 20

```
  cout << "Value at p: " << *p << endl;
  cout << "Value of x: " << x << endl;
```

prints 7

Will reveal in the next figure

```
}
```

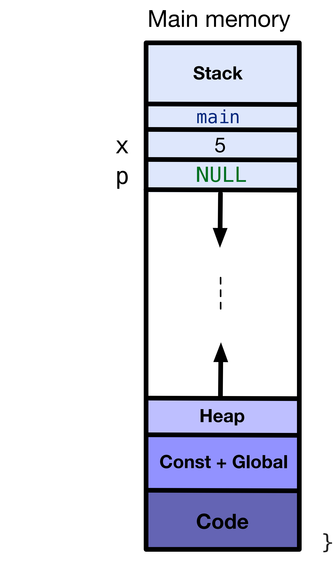



```
#include <iostream>
using namespace std;

int main(){
  int x = 10;
  int* p = NULL;
  p = &x;
  *p = 5;

  p = new int;



  *p = 20;

  cout << "Value at p: " << *p << endl;
  cout << "Value of x: " << x << endl;


  delete p;
```

free address stored in p (the heap address)

Will reveal in the next figure

```
}
```




```
#include <iostream>
using namespace std;

int main(){
  int x = 10;
  int* p = NULL;
  p = &x;
  *p = 5;

  p = new int;



  *p = 20;

  cout << "Value at p: " << *p << endl;
  cout << "Value of x: " << x << endl;


  delete p;
  p = NULL;
  return 0;
}
```