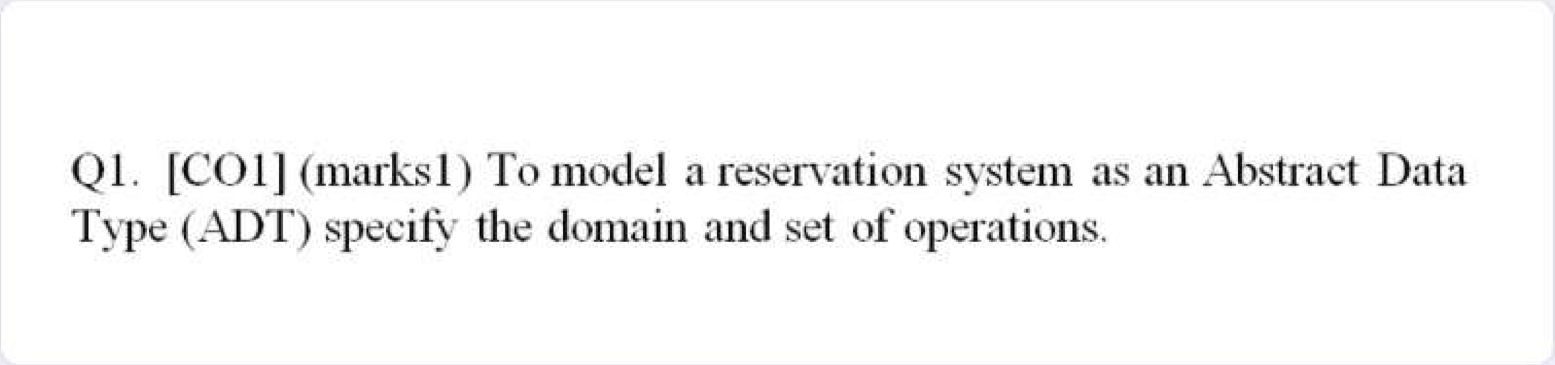
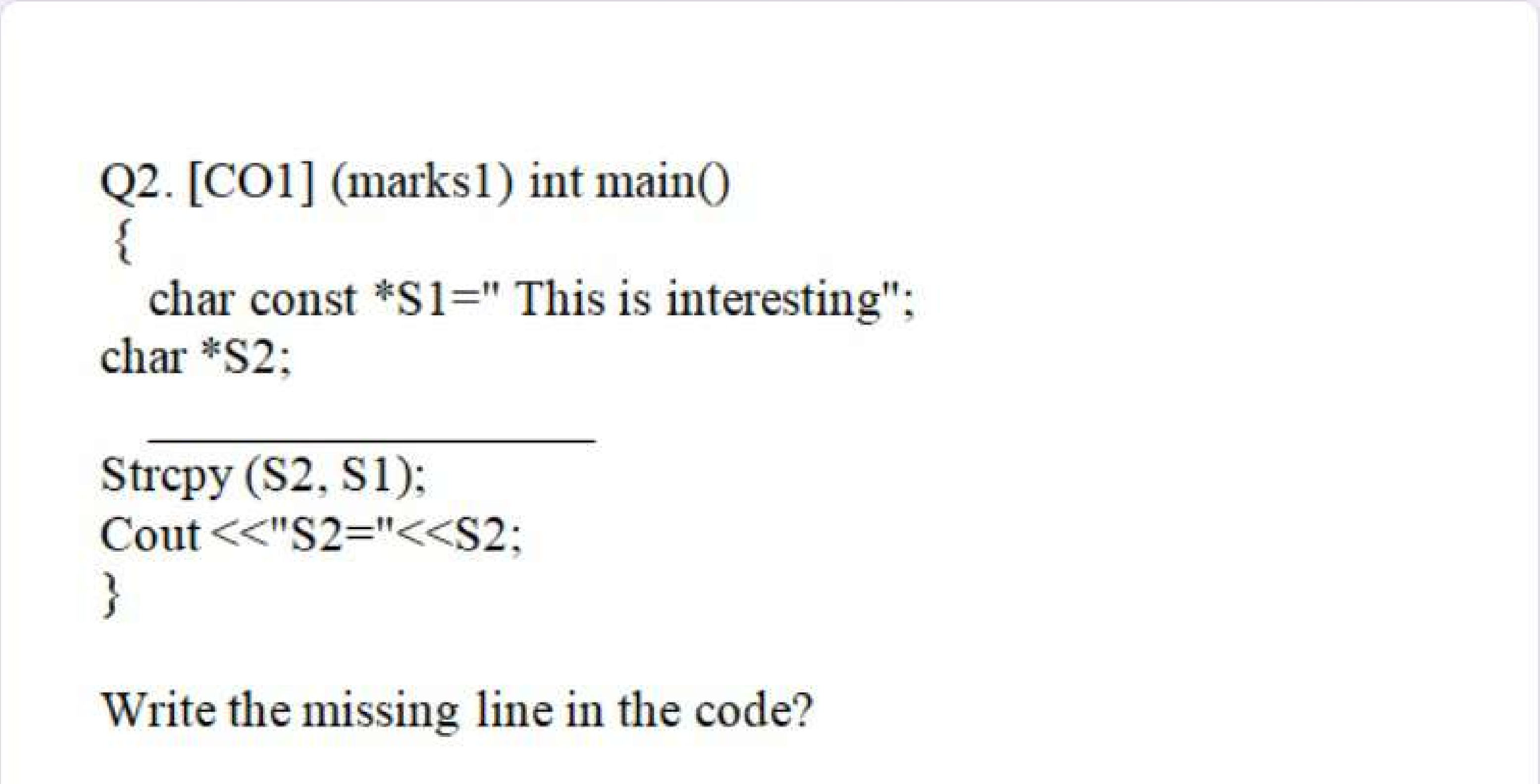
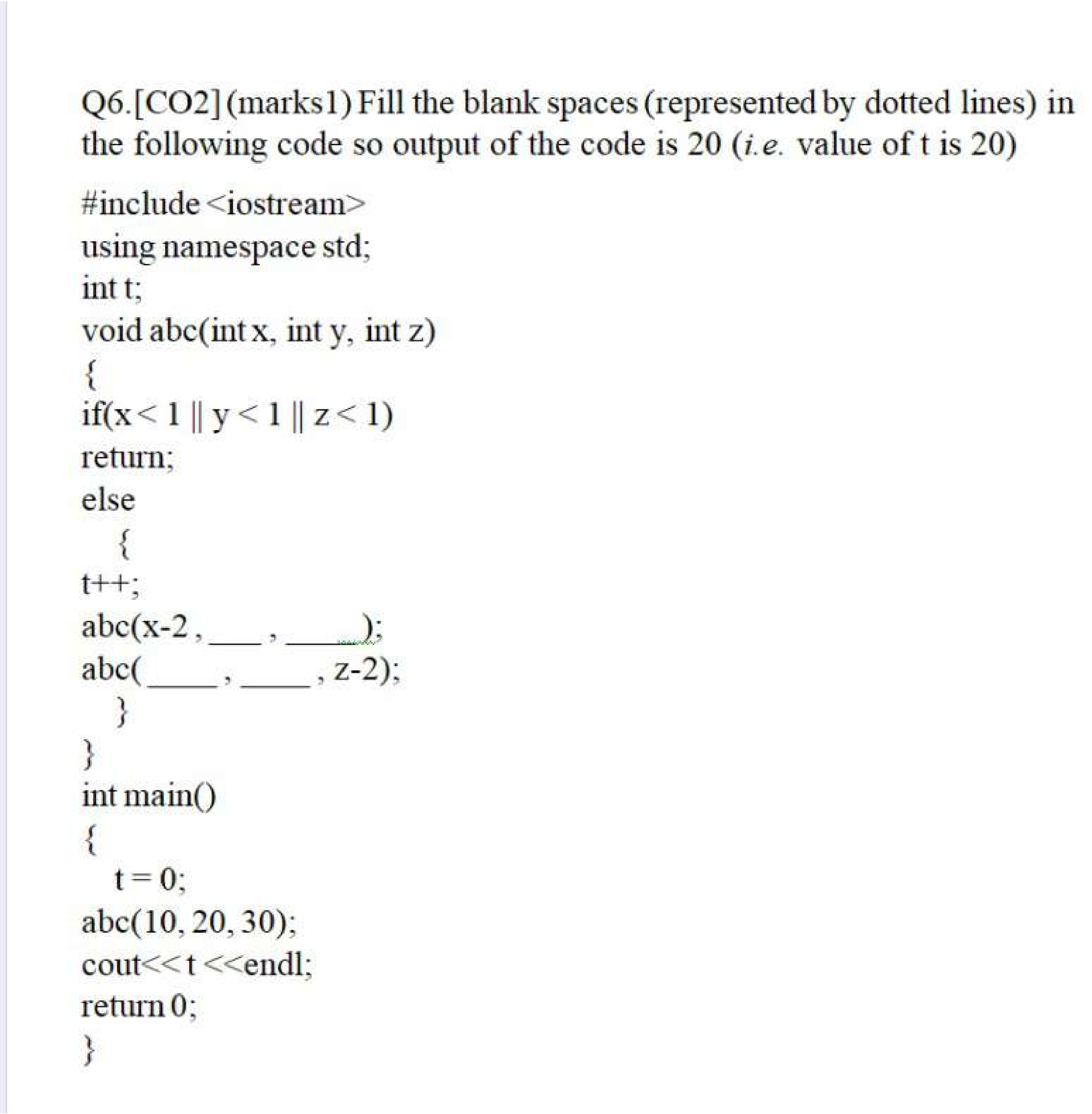


Q1. [CO1] (marks1) To model a reservation system as an Abstract Data Type (ADT) specify the domain and set of operations.



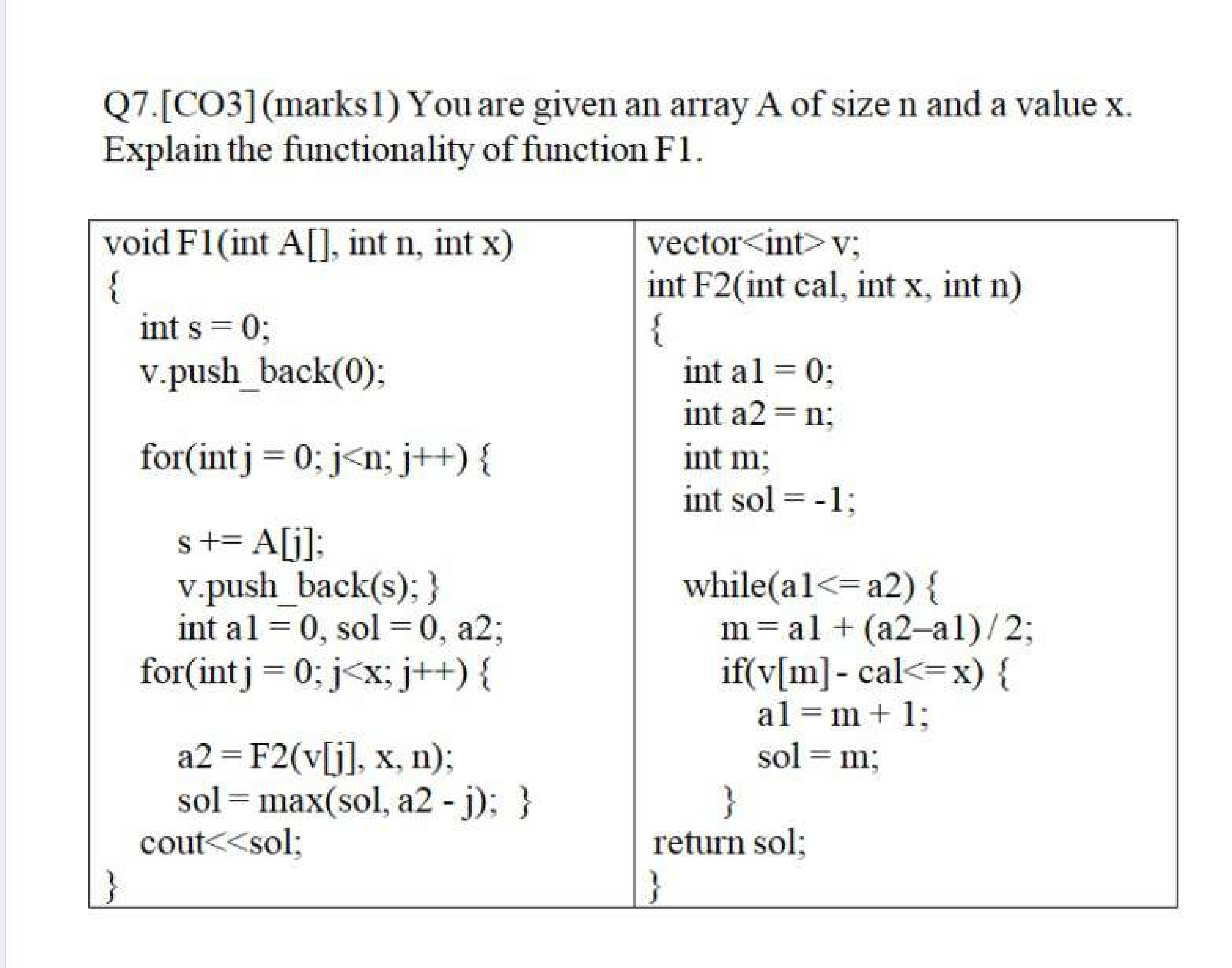
```
Q2. [CO1] (marks1) int main()
{
   char const *S1=" This is interesting";
char *S2;

   ____________________
Strcpy (S2, S1);
Cout <<"S2="<<S2;
}
```

Write the missing line in the code?



Q6.[CO2] (marks1) Fill the blank spaces (represented by dotted lines) in the following code so output of the code is 20 (*i.e.* value of t is 20)

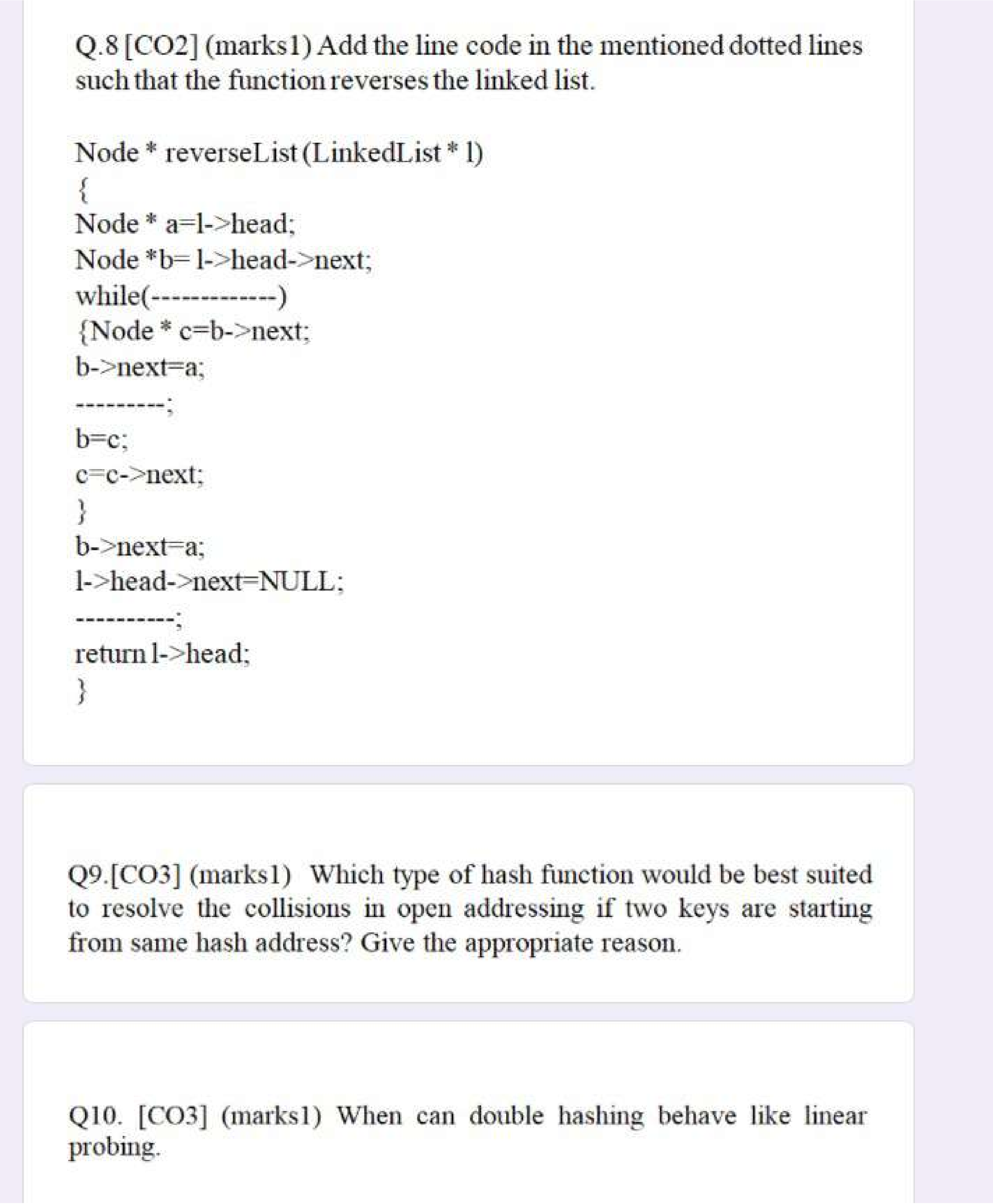
```
#include <iostream>
using namespace std;
int t;
void abc(int x, int y, int z)
{
if(x < 1 || y < 1 || z < 1)
return;
else
   {
t++;
abc(x-2 , ___ , _____);
abc( _____ , _____ , z-2);
   }
}
int main()
{
   t = 0;
abc(10, 20, 30);
cout<< t <<endl;
return 0;
}
```



Q7.[CO3] (marks1) You are given an array A of size n and a value x. Explain the functionality of function F1.

```
void F1(int A[], int n, int x)
{
   int s = 0;
   v.push_back(0);

   for(int j = 0; j<n; j++) {

      s += A[j];
      v.push_back(s); }
      int a1 = 0, sol = 0, a2;
   for(int j = 0; j<x; j++) {

      a2 = F2(v[j], x, n);
      sol = max(sol, a2 - j);  }
   cout<<sol;
}
```

```
vector<int> v;
int F2(int cal, int x, int n)
{
   int a1 = 0;
   int a2 = n;
   int m;
   int sol = -1;

   while(a1<= a2) {
      m = a1 + (a2–a1) / 2;
      if(v[m] - cal<= x) {
         a1 = m + 1;
         sol = m;
      }
 return sol;
}
```



Q.8 [CO2] (marks1) Add the line code in the mentioned dotted lines such that the function reverses the linked list.

```
Node * reverseList (LinkedList * l)
{
Node * a=l->head;
Node *b= l->head->next;
while(-------------)
{Node * c=b->next;
b->next=a;
---------;
b=c;
c=c->next;
}
b->next=a;
l->head->next=NULL;
----------;
return l->head;
}
```

Q9.[CO3] (marks1) Which type of hash function would be best suited to resolve the collisions in open addressing if two keys are starting from same hash address? Give the appropriate reason.

Q10. [CO3] (marks1) When can double hashing behave like linear probing.