```
(penalty regime, 0 70)
     #include<stdio.h>
  2 v int main(){
          int t;
  3
          scanf("%d",&t);
  4
         while(t--){
  5 ▼
              int n;
  6
              scanf("%d",&n);
  7
              int a[n];
 8
              for(int i=0;i<n;i++){
 9 🔻
                   scanf("%d",&a[i])
10
11
              int k;
12
              scanf("%d",&k);
13
              int flag=0;
14
              for(int i=0;i<n;i++){
15 ▼
                   for(int j=i+1;j<n</pre>
16 ▼
                       if(a[i]-a[j]=
17
18
              if(flag) break;}
19
              printf("%d\n",flag);
20
21
    }
22
```

	Input	Expected	Got	
~	1 3 1 3 5 4	1	1	~
~	1 3 1 3 5 99	0	0	~

Passed all tests! <

Sam buys 1 chocolate on day 1, 0 on day 2, and 3 on day 3. This gives us a total of 4 chocolates. Thus, we print 4 on a new line.

Answer: (penalty regime: 0 %)

```
#include<stdio.h>
  1
  2 ▼
     int main(){
 3
         int t;
         scanf("%d",&t);
 4
 5 ▼
         while(t--){
 6
             int n,c=0;
 7
             scanf("%d",&n);
 8 🔻
             for(int i=0;i<=n;i++)
                  if(i%2!=0) c=c+i;
 9
10
             }printf("%d\n",c);
11
12
   }
13
```

	Input	Expected	Got	
~	3	1	1	~
	1	1	1	
	2	4	4	
	3			

	Input	Expected	Got	
~	3	1	1	~
	1	1	1	
	2	4	4	
	3			
~	10	1296	1296	~
	71	2500	2500	
	100	1849	1849	
	86	729	729	
	54	400	400	
	40	25	25	
	9	1521	1521	
	77	25	25	
	9	49	49	
	13	2401	2401	
	98			

Passed all tests! <

```
#include<stdio.h>
 1
    int main()
 2
 3 √ {
         int s1,s2,ans;
 4
         scanf("%d",&s1);
 5
         int ta[s1];
 6
         for(int i=0;i<s1;i++)
 7
         scanf("%d",&ta[i]);
 8
         scanf("%d",&s2);
 9
         int tb[s2];
10
         for(int i=0;i<s2;i++)</pre>
11
         scanf("%d",&tb[i]);
12
         for(int j=0;j<s2;j++)</pre>
13
         {
14 ▼
             ans=0;
15
             for(int i=0;i<s1;i++)
16 ▼
                  if(tb[j]>=ta[i])
17
                  ans++;
18
             }printf("%d\n",ans);
19
20
   |}
21
```

	Input	Expected	Got	
/	4	2	2	~
	1	4	4	
	4			
	2			
	4			
	2			
	3			
	5			
	5	1	1	~
•	2	0	0	
	10	2	3	

```
#include<stdio.h>
   1
      int main(){
   2 v
          int t;
  3
          scanf("%d",&t);
  4
          while(t--){
  5 ▼
              int n;
  6
              scanf("%d",&n);
  7
              int a[n];
  8
              for(int i=0;i<n;i++){
  9 ▼
                   scanf("%d",&a[i])
 10
 11
              int k;
 12
              scanf("%d",&k);
13
              int flag=0;
14
              for(int i=0;i<n;i++){
15 ▼
                  for(int j=i+1;j<n
16 ▼
                       if(a[i]-a[j]=
17
18
              if(flag) break;}
19
              printf("%d\n",flag);
20
21
   }
22
```

	Input	Expected	Got	
~	1 3 1 3 5 4	1	1	~
~	1 3 1 3 5 99	0	0	~

Passed all tests! ✓

chocolates. Thus, we print 4 on a new line.

Answer: (penalty regime: 0 %)

```
#include<stdio.h>
 1
    int main(){
 3
        int t;
 4
        scanf("%d",&t);
 5 ▼
        while(t--){
 6
             int n,c=0;
 7
             scanf("%d",&n);
 8 🔻
             for(int i=0;i<=n;i++)
 9
                 if(i%2!=0) c=c+i;
10
             }printf("%d\n",c);
11
12
13
```

	Input	Expected	Got	
~	3	1	1	~
	1	1	1	
	2	4	4	
	3			
~	10	1296	1296	~
	71	2500	2500	
	100	1849	1849	
	86	729	729	
	54	400	400	
	40	25	25	
	9	1521	1521	
	77	25	25	
	9	49	49	
	13	2401	2401	
	98			

Passed all tests! 🗸

Answer: (penalty regime: 0 %)

```
#include<stdio.h>
  2
      int main()
  3 √ {
  4
          int s1,s2,ans;
  5
          scanf("%d",&s1);
  6
          int ta[s1];
          for(int i=0; i<s1; i++)
  7
          scanf("%d",&ta[i]);
  8
          scanf("%d",&s2);
  9
          int tb[s2];
 10
          for(int i=0;i<s2;i++)</pre>
11
          scanf("%d",&tb[i]);
12
          for(int j=0; j<s2; j++)
13
14 ▼
         {
15
              ans=0;
              for(int i=0;i<s1;i++)
16 ▼
                  if(tb[j]>=ta[i])
17
18
                  ans++;
              }printf("%d\n",ans);
19
20
21
```

	Input	Expected	Got	
~	4	2	2	~
	1	4	4	
	4			
	2			
	4			
	2			
	3			
	5			
~	5	1	1	~
	2	0	0	
	10	3	3	
	5	4	4	
	4			
	8			
	4 3			
	3			
	1			
	7			
	8			