```c
#include<stdio.h>
int main(){
    int a,b;
    scanf("%d",&a);
    int arr1[a];
    for(int i=0;i<a;i++)
    scanf("%d",&arr1[i]);
    scanf("%d",&b);
    int arr2[b];
    for(int i=0;i<b;i++)
    scanf("%d",&arr2[i]);
    int p=0,q=0;
    while((p<a)&&(q<b)){
        if(arr1[p]<arr2[q]){
            printf("%d ",arr1[p]);
            p++;
        }
        else if(arr1[p]>arr2[q]){
            printf("%d ",arr2[q]);
            q++;
        }
        else
        {
            printf("%d ",arr1[p]);
            p++;
            q++;
        }
    }
    for(int j=p;j<a;j++){
        printf("%d ",arr1[j]);
    }
    for(int j=q;j<b;j++){
        printf("%d ",arr2[j]);
    }

}
```

	Input	Expected	Got	
✓	5 1 2 3 6 9 4 2 4 5 10	1 2 3 4 5 6 9 10	1 2 3 4 5 6 9 10	✓

Answer: (penalty regime: 0 %)

```c
#include<stdio.h>
int main ()
{
    int t;
    scanf("%d",&t);
    while(t--){
        int n;
        scanf("%d",&n);
        int a[n];
        for(int i=0;i<n;i++){
            scanf("%d",&a[i]);
        }
        int k;
        scanf("%d",&k);
        int flag = 0;
        for(int i=0;i<n;i++){
            for(int j=i+1;j<n;j++){
                if(a[i] - a[j] == k||a[j] - a[i] == k)
                {
                    flag = 1;
                    break;
                }
            }
            if(flag)
            break;}
            printf("%d\n",flag);
        }
    return 0;
}
```

	Input	Expected	Got	
✓	1 3 1 3 5 4	1	1	✓
✓	1	0	0	✓

```c
#include<stdio.h>
int main ()
{
    int t;
    scanf("%d",&t);
    while(t--){
        int n,c=0;
        scanf("%d",&n);
        for(int i =0;i<=n;i++){
            if(i%2!=0)
            c = c+i;
        }
        printf("%d\n",c);
    }
}
```

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

Answer: (penalty regime: 0 %)

```c
#include<stdio.h>
int main ()
{
    int s1,s2,ans;
    scanf("%d",&s1);
    int ta[s1];
    for(int i=0;i<s1;i++)
    scanf("%d",&ta[i]);
    scanf("%d",&s2);
    int tb[s2];
    for(int i=0;i<s2;i++)
    scanf("%d",&tb[i]);
    for(int j=0;j<s2;j++){
        ans = 0;
        for(int i=0;i<s1;i++){
            if(tb [j] >=ta[i])
            ans ++;
        }
        printf("%d\n",ans);
    }
}
```

	Input	Expected	Got	
✓	4 1 4 2 4 2 3 5	2 4	2 4	✓
✓	5 2 10 5	1 0 3 4	1 0 3 4	✓

2 x 10 = 20

Answer: (penalty regime: 0 %)

```c
#include<stdio.h>
int main (){
    int n;
    scanf("%d",&n);
    for(int i=1;i<=10;i++){
        printf("%d x %d = %d\n",n,i,n*i);
    }
}
```

	Input	Expected	Got	
✓	2	2 x 1 = 2 2 x 2 = 4 2 x 3 = 6 2 x 4 = 8 2 x 5 = 10 2 x 6 = 12 2 x 7 = 14 2 x 8 = 16 2 x 9 = 18 2 x 10 = 20	2 x 1 = 2 2 x 2 = 4 2 x 3 = 6 2 x 4 = 8 2 x 5 = 10 2 x 6 = 12 2 x 7 = 14 2 x 8 = 16 2 x 9 = 18 2 x 10 = 20	✓

Passed all tests! ✓

Explanation 2

2 + 3 = 5, is the best case for maximum nutrients.

Answer: (penalty regime: 0 %)

```c
#include<stdio.h>
int main(){
    long long int n,t,i,nut = 0;
    scanf("%lld %lld",&n,&t);
    for(i=1;i<=n;i++){
        nut = nut+i;
        if(nut==t){
            nut =nut-1;
        }
    }
    printf("%lld",nut%1000000007);
}
```

	Input	Expected	Got	
✓	2 2	3	3	✓
✓	2 1	2	2	✓
✓	3 3	5	5	✓

Passed all tests! ✓

1

Explanation 2

Factoring $n = 1$ we get $\{1\}$. We then return the $p = 1^{st}$ factor as our answer.

Answer: (penalty regime: 0 %)

```c
#include<stdio.h>
long f(int n,int p){
    int factors[1000];
    int count = 0;
    for(int i=1;i<=n;i++){
        if(n%i==0){
            factors[count++]=i;
        }
    }
    if(p>count){
        return 0;
    }
    return factors[p-1];
}
int main (){
    int n,p;
    scanf("%d",&n);
    scanf("%d",&p);
    printf("%ld\n",f(n,p));
    return 0;
}
```

	Input	Expected	Got	
✓	10 3	5	5	✓
✓	10 5	0	0	✓
✓	1 1	1	1	✓

Passed all tests! ✓

```c
#include<stdio.h>
int main (){
    int n,k;
    scanf("%d",&n);
    int arr[n];
    for(int i=0;i<n;i++){
        scanf("%d",&arr[i]);
    }
    scanf("%d",&k);
    for(int a=0;a<=n-k;a++){
        int max=arr[a];
        for(int b=a;b<a+k;b++){
            if(arr[b]>max){
                max=arr[b];
            }
        }
        printf("%d ",max);
    }
}
```

	Input	Expected	Got	
✓	8 1 3 5 2 1 8 6 9 3	5 5 5 8 8 9	5 5 5 8 8 9	✓
✓	10 3 7 5 1 2 9 8 5 3 2	7 7 5 9 9 9 8 5	7 7 5 9 9 9 8 5	✓

6 5 8 10 13 6 2 3	17
7 20 35 57 30 56 87 30 10	33

Answer: (penalty regime: 0 %)

```c
#include<stdio.h>
int main(){
    int n,t,count = 0;
    scanf("%d",&n);
    int arr[n];
    for(int i=0;i<n;i++){
        scanf("%d",&arr[i]);
    }
    scanf("%d",&t);
    for(int j=0;j<n;j++)
    {
        while(arr[j]>0){
            arr[j]=t;count++;
        }
    }
    printf("%d",count);
}
```