

Given two numbers, write a C program to swap the given numbers.

For example:

Input	Result
10 20	20 10

Answer: (penalty regime: 0 %)

```
1 #include<stdio.h>
2 int main(){
3     int a;
4     int b;
5     scanf("%d %d",&a,&b);
6     printf("%d %d",b,a);
7     return 0;
8 }
```

	Input	Expected	Got	
✓	10 20	20 10	20 10	✓

Passed all tests! ✓

Correct

Marks in Maths == 65
Marks in Physics == 85
Marks in Chemistry == 50
Or
Total in all three subjects == 100

Sample Test Cases

Test Case 1

Input

70 80 80

Output

The candidate is eligible

Test Case 2

Input

60 80 80

Output

The candidate is eligible

Test Case 3

Input

50 60 40

Output

The candidate is not eligible

Answer: (partial score: 0.8)

```
1: //Program to check if candidate is eligible or not
2: int main()
3: {
4:     int marks[3];
5:     cout << "Enter marks in Maths, Physics and Chemistry : ";
6:     cin >> marks[0] >> marks[1] >> marks[2];
7:     int totalmarks;
8:     totalmarks = marks[0] + marks[1] + marks[2];
9:     if(totalmarks >= 100)
10:    {
11:        cout << "The candidate is eligible";
12:    }
13:    else
14:    {
15:        cout << "The candidate is not eligible";
16:    }
17:    return 0;
18: }
```

Input	Expected	Got
70 80 80	The candidate is eligible	The candidate is eligible ✓
60 80 80	The candidate is eligible	The candidate is eligible ✓

Passed all tests! ✓

Grade for this submission: 100.00%



Sabu is very kind to beggars and every day Sabu donates half of the amount he has when ever a beggar requests him. The money M left in Sabu's hand is passed as the input and the number of beggars B who received the alms are passed as the input. The program must print the money Sabu had in the beginning of the day.

Input Format:

The first line denotes the value of M.
The second line denotes the value of B.

Output Format:

The first line denotes the value of money with Sabu in the beginning of the day.

Example Input/Output:

Input:

100
2

Output:

400

Explanation:

Sabu donated to two beggars. So when he encountered second beggar he had $100 \times 2 = 100$, 200 and when he encountered 3rd he had $200 \times 2 = 100$, 400.

Answer: (penalty regime 0 %)

```
1 //Programm Sabu.h
2 int main(){
3     int M;
4     int B;
5     int A;
6     scanf("%d %d", &M, &B);
7     A=M*B;
8     printf("%d", A);
9     return 0;
10 }
```

Input	Expected	Got
100 2	100 400	100 400

Passed all testcases ✓

Correct

Starts for this submission | 2021.30

◀ Previous page

Next page ▶

The CEO of company ABC inc. wanted to encourage the employees coming on time to the office. So he announced that for every consecutive day an employee comes on time in a week (starting from Monday to Saturday) he will be awarded Rs.200 more than the previous day as "Punctuality incentive". The input : for the starting day (x on Monday) is passed as the input to the program. the number of days N an employee came on time consecutively starting from Monday is also passed as the input. The program must calculate and print the "Punctuality incentive" of the employee.

Input Format:

The first line denotes the value of x .
The second line denotes the value of N .

Output Format:

The first line denotes the value of P .

Example Input/Output:

Input:

500
3

Output:

2100

Explanation:

On Monday the employee receives Rs.500, on Tuesday Rs.700, on Wednesday Rs.900

So total = Rs.2100

Answer: (penalty regime: 0.8)

```
1 #include<iostream.h>
2 int main()
3 {
4     int x,M,amount=0;
5     cin>>x>>M;
6     for(i=x;i<=M;i++)
7     {
8         amount+=i;
9         i+=20;
10    }
11    cout<<"Rs."<<amount;
12    return 0;
13 }
```

Input	Expected	Got
✓ 500 3 x M	2100	2100 ✓
✓ 500 5 x M	1600	1600 ✓

Passed all tests! ✓

Score:

Score for this submission: 1.00/1.00

[Previous page](#)

[Next page](#) ▶

Two matrices M and N are passed as the input. A number X is also passed as the input. The program must print the numbers divisible by X from N to M (inclusive of M and N).

Input Format:

The first line denotes the value of M.
The second line denotes the value of N.
The third line denotes the value of X.

Output Format:

Numbers divisible by X from N to M, with each number separated by a space.

Boundary Conditions:

1 <= M <= 999999
M <= N <= 999999
1 <= X <= 999

Example Input Output 1:

Input:
2
40
7

Output:
33 28 21 14 7

Example Input Output 2:

Input:
66
121
11

Output:
121 110 99 88 77 66

Answer: (partial score: 0 %)

```
1 // recursive solution
2 int max();
3 {
4     int M;
5     int N;
6     int X;
7     scanf("%d %d %d", &M, &N, &X);
8     for (int i = M; i <= N; i++)
9     {
10         if (i % X == 0)
11             printf("%d ", i);
12     }
13 }
14 return 0;
15 }
```

Input	Expected	Got	
2 40 7	33 28 21 14 7	33 28 21 14 7	✓

Passed all tests! ✓

Done

Works for 75 submissions | 1.00% AC

Previous page

Next page ►

Two numbers M and N are passed as the input. A number X is also passed as the input. The program must print the numbers divisible by X from N to M (inclusive of M and N).

Input Format:

The first line denotes the value of M.
The second line denotes the value of N.
The third line denotes the value of X.

Output Format:

Numbers divisible by X from N to M, with each number separated by a space.

Boundary Conditions:

1 <= M <= 999999
M < N < 999999
1 <= X <= 999

Example Input Output 1:

Input:
2
40
7

Output:
35 28 21 14 7

Example Input Output 2:

Input:
60
120
11

Output:
120 110 99 88 77 66

Answer: (penalty regime: 0 %)

```
1 //Program to print all numbers divisible by X from N to M.
2 int main()
3 {
4     int M;
5     int N;
6     int X;
7     int R;
8
9     scanf("%d %d %d", &M, &N, &X);
10    for (int i=M;i>=N;i--)
11    {
12        if (i%X==0)
13            printf("%d ", i);
14    }
15    return 0;
16 }
```

Input	Expected	Got	
✓ 2 40 7	35 28 21 14 7	35 28 21 14 7	✓

Passed all tests! ✓

Correct

Wait for the submission (100/100).

Write a C program to find the quotient and remainder of given integers.

For example:

Input	Result
12	1
1	0

Answer: (penalty regime: 0 %)

```
1 //includestdio.h
2 int main(){
3     int a,b;
4     scanf("%d %d", &a, &b);
5     int c,a/b;
6     int d=a%b;
7     printf("%d%d",c,d);
8     return 0;
9 }
```

	Input	Expected	Get
✓	12	1	1
	1	0	0

Passed all tests! ✓

Correct

Marks for this submission: 1.00 / 1.00

Write a C program to find the biggest among the given 3 integers!

For example:

Input	Result
10 20 30	30

Answer: (penalty regions: 0.0)

```
1 #include<stdio.h>
2 int main(){
3     int a,b,c;
4     scanf("%d %d %d", &a,&b,&c);
5     if((a+b+c) > c){
6         printf("%d", a);
7     }
8     else if ((b+c) > a){
9         printf("%d", b);
10    }
11    else{
12        printf("%d", c);
13    }
14    return 0;
15 }
16
17
```

Input	Expected	Got
✓ 10 20 30	30	30 ✓

Passed all tests! ✓

Details

Marks for this submission: 1.0/1.0.

◀ Previous page

Next page ▶

Write a C program to find whether the given integer is odd or even?

For example:

Input	Result
12	Even
11	Odd

Answer: (penalty regime: 0 %)

```
1 //EvenOrOdd.catalina.java
2
3 int main(){
4     int a;
5     Scanner s = new Scanner(System.in);
6     a = s.nextInt();
7     if(a%2 == 0){
8         System.out.println("Even");
9     }
10    else{
11        System.out.println("Odd");
12    }
13 }
```

Input	Expected	Got
12	Even	Even ✓
11	Odd	Odd ✓

Passed all tests! ✓

Correct

Status for this submission: 1.00/1.00.

◀ Previous page

Write a C program to find the factorial of given n.

For example:

Input	Result
5	120

Answer: (penalty: 0 h)

```
1 #include<stdio.h>
2 int main(){
3     int n;
4     int fact = 1;
5     scanf("%d",&n);
6     for (int i=1;i<=n;i++){
7         fact *= i;
8     }
9     printf("%d",fact);
10 }
```

Input	Expected	Get
5	120	120 ✓

Passed all test! ✓

Done

Mark for this submission: 1.00/1.00

◀ Previous page

Next page ▶

Write a C program to find the sum first N natural numbers.

For example:

Input	Result
5	15

Answer: (penalty regime: 0.16)

```
1 #include<stdio.h>
2 + int main()
3 { int n,sum;
4   scanf("%d",&n);
5   for(int i=1;i<=n;i++)
6   {
7     sum+=i;
8   }
9   printf("%d",sum);
10 }
```

Input	Expected	Got
5	15	15 ✓

(Total attempts: 1 ✓)

Correct

Marks for this submission: 1.00/1.00

◀ Previous page

Next page ▶

Write a C program to find the nth term in the Fibonacci series.

For example:

Input	Result
0	0
1	1
2	1

Answer: (penalty region: 0 %)

```
1 #include<stdio.h>
2 int main(){
3     int a=0,b=1,c;
4     scanf("%d",&n);
5     if(n==0){
6         printf("0");
7     }
8     else if(n==1){
9         printf("1");
10    }
11    else{
12        for(int i=1;i<n;i++){
13            c=a+b;
14            a=b;
15            b=c;
16        }
17        printf("%d", b);
18    }
19 }
```

Input	Expected	Got
0	0	0 ✓
1	1	1 ✓
2	1	1 ✓

Passed all trials! ✓

Correct

Score for this submission: 1.00/1.00

Previous page

Next page ►

Write a C program to find the power of integers.

Input:

a b

Output:

a^b value

For example:

Input	Result
3 4	81

Answer: (penalty: 0 mins, 0 %)

```
1 #include<stdio.h>
2 #include<math.h>
3 int main()
4 {
5     int a,b;
6     scanf("%d %d",&a,&b);
7     printf("%d",pow(a,b));
}
```

Input	Expected	Got
✓ 2 4	16	16 ✓

Passed at least: ✓

Submitted

Marks for this submission: 1.00/1.00

[Previous page](#)

[Next page](#) ▶

Write a C program to find whether the given integer is prime or not.

For example:

Input	Result
7	prime
8	Not prime

Answer: (penalty: 0%)

```
1 //include<stdio.h>
2 int main(){
3     int n,flag=0;
4     scanf("%d",&n);
5
6     if (n==1||n<1){
7         flag=1;
8     }
9     else{
10        for(int i=2;i<n;i+=1){
11            if(n% i==0){
12                flag=1;
13                break;
14            }
15        }
16    }
17    if(flag==0){
18        printf("prime");
19    }
20    else{
21        printf("Not prime");
22    }
23 }
```

Input	Expected	Got
7	prime	prime ✓
8	Not prime	Not prime ✓

Passed all tests! ✓

Correct

Mark for this submission: 1.00/1.00

[Previous page](#)



[Next page](#) ➔

Write a C program to find the reverse of the given integer?

Answer: (penalty regime: 0.0)

```
1 //reverseint.c
2
3 int main(){
4     int a,reversed=0,remainder;
5     scanf("%d", &a);
6     while(a!=0){
7         remainder=a%10;
8         reversed=reversed*10+remainder;
9         a/=10;
10    }
11    printf("%d",reversed);
12 }
```

Input	Expected	Get
✓ 123	321	321 ✓

Passed all tests! ✓

Correct

Mark for this submission: 1.00/1.0.

Question 1 | Score: Mark 1.00 out of 1.00 [View question](#)

Convert the following algorithm into a program and find its time complexity using the counter method.

```
void function (int n)
{
    int k = 1;
    int s = 1;
    while(s <= n)
    {
        k++;
        s += k;
    }
}
```

Note: No need of counter increment for declarations and `scanf()` and `cout` variable `printf()` statements.

Input:
A positive integer `n`.
Output:
Print the value of the counter variable.

For example:

Input	Result
8	12

Answer: (parity regime: 0 %)

```
1 #include <iostream.h>
2
3 void function (int n)
4 {
5     int counter = 0;
6     int k = 1;
7     int s = 1;
8     counter += 2;
9
10    while(s <= n)
11    {
12        counter++;
13        k++;
14        counter++;
15        s += k;
16        counter++;
17    }
18    counter++;
19
20    printf("%d\n", counter);
21 }
22
23 int main()
24 {
25     int n;
26     scanf("%d", &n);
27     function(n);
28     return 0;
29 }
```

Input	Expected	Got
8	12	12 ✓
10	16	16 ✓

Passed all tests! ✓

Correct

Mark for this submission: 1.00/1.00.

```

convert the following algorithm into a program and find its time complexity using the counter method.

void func(int n)
{
    if(n==1)
    {
        printf("*");
    }
    else
    {
        for(int i=1; i<n; i++)
        {
            for(int j=i; j<n; j++)
            {
                printf("*");
                printf("*");
                break;
            }
        }
    }
}

```

Note: No need of counter increment for declarations and sizeof() nor count variable printf() statements.
Input:
A positive integer n .
Output:
Print the value of the counter variable.

Answer: (penalty regimen: 0 %)

```

1. #include <stdio.h>
2.
3. void func(int n) {
4.     long long counter = 0;
5.
6.     if (n == 1) {
7.         counter += 2;
8.     } else {
9.         counter = 5LL * n + 2;
10.    }
11.
12.    printf("%lld\n", counter);
13. }
14.
15. int main() {
16.     int n;
17.     scanf("%d", &n);
18.     func(n);
19.     return 0;
20. }

```

	Input	Expected	Got	
✓	2	12	12	✓
✓	1000	1002	1002	✓
✓	100	712	712	✓

Passed all tests! ✓

Correct

Mark for this submission: 1.00/1.00.

Convert the following algorithm into a program and find its time complexity using master method.

```

factor(num) {
{
    for (i = 1; i < num; i++)
    {
        if (num % i == 0)
        {
            printf("%d ", i);
        }
    }
}

```

Note: No need of counter increment for declarations and `scanf()` and `cout` variable `printf()` statement.

Input:

A positive integer n .

Output:

Print the value of the counter variable.

Answer:

```

1 #include <iostream.h>
2
3 int main() {
4     int num, c = 0;
5     scanf("%d", &num);
6
7     for (int i = 1; i < num; i++) {
8         c++;
9         c++;
10
11         if (num % i == 0) {
12             c++;
13         }
14     }
15
16     c++;
17     printf("%d", c);
18     return 0;
19 }
20

```

	Input	Expected	Got	
✓	12	6	6	✓
✓	25	10	10	✓
✓	4	2	2	✓

Passed all test! ✓

Correct

Marks for this subexercise: 1.00/1.00.

Convert the following algorithm into a program and find its time complexity using counter method.

```

void function(int n)
{
    int c = 0;
    for(int i=n/2; i>n; i--)
        for(int j=1; j<n; j+= 2 * i)
            for(int k=i; k<n; k+= i * i)
                i++;
}

NOTE: no need of counter increment for declarations and cout() and count variable print() statements.

Input:  

A positive integer n  

Output:  

Print the value of the counter variable

```

Answer:

```

1 #include <iostream.h>
2
3 int main() {
4     int n, count = 0;
5     scanf("%d", &n);
6     int c = 0;
7     count++;
8
9     for (int i = n / 2; i < n; i++) {
10         count++;
11         for (int j = 1; j < n; j += 2 * i) {
12             count++;
13             for (int k = i; k < n; k += i * i) {
14                 count++;
15                 c++;
16                 count++;
17             }
18         }
19         count++;
20     }
21     count++;
22     printf("%d", count);
23     return 0;
24 }
25

```

	Input	Expected	Got	
✓	1	0	0	✓
✓	10	212	212	✓

Passed all tests! ✓

Correct

Mark for this question: 1.00/1.00

Convert the following algorithm into a program and find its time complexity using master method.

```
void reverse(int n)
{
    int rev = 0, remainder;
    while (n != 0)
    {
        remainder = n % 10;
        rev = rev * 10 + remainder;
        n /= 10;
    }
    printf("%d");
}
```

Note: No need of counter increment for declarations and `scanf()` and `cout` variable `printf()` statements.

Input:
A positive integer `n`
Output:
Print the value of the counter variable

Answer:

```
1 #include<cs50.h>
2
3 int main() {
4     int n,c=0;
5     scanf("%d",&n);
6     int rev=0,remainder;
7     while(n!=0) { c++;
8         remainder=n%10;
9         rev=rev*10+remainder;
10        n/=10;
11    } c++;
12    cout<<c;
13    return 0;
14 }
```

	Input	Expected	Got	
✓	11	11	11	✓
✓	1234	19	19	✓

Passed all test! ✓

Correct

Marks for this submission: 1.00/1.00.

Practical Statement
Given an array of 'n' and 'm' blocks of 'k' and following all the rules, is to find the number of 'k's. Write a program using Bubble and Dijkstra to Count the number of zeros in the given array.
Input Format
First Line Contains Integer - Size of array
Second Line Contains m numbers of an array
Output Format
First Line Contains Integer - Number of zeros present in the given array.

Zusammenfassung (Fiktiv) 10 %

```

class Solution {
public:
    int searchRange(vector<int>& nums, int low, int high, int value) {
        if (low > high)
            return -1;
        int mid = (low + high) / 2;
        if (value == nums[mid])
            if (mid == 0 || value < nums[mid-1])
                return mid;
            else
                return searchRange(nums, low, mid-1, value);
        else if (value < nums[mid])
            return searchRange(nums, low, mid-1, value);
        else
            return searchRange(nums, mid+1, high, value);
    }
};

int main() {
    Solution sol;
    cout << sol.searchRange({5,7,7,8,8,10}, 0, 5, 8);
    cout << endl;
    cout << sol.searchRange({5,7,7,8,8,10}, 0, 5, 6);
    cout << endl;
    cout << sol.searchRange({5,7,7,8,8,10}, 0, 5, 10);
    cout << endl;
    cout << sol.searchRange({5,7,7,8,8,10}, 0, 5, 11);
    cout << endl;
}

```

Given an array `nums` of size n , return the majority element.

The majority element is the element that appears more than $\lfloor n / 2 \rfloor$ times. You may assume that the majority element always exists in the array.

Example 1:

Input: `nums` = [2,2,1,1,1,2,2]

Output: 1

Example 2:

Input: `nums` = [2,2,1,1,1,2,2,2]

Output: 2

Constraints:

- $n == \text{nums.length}$
- $1 \leq n \leq 5 * 10^4$
- $-2^{31} \leq \text{nums}[i] \leq 2^{31} - 1$

For example:

Input	Result
2	2
2 2 2	
2	2
2 2 1 1 1 2 2	

Answer: (penalty regime: 0 %)

```
1 #include <stdio.h>
2
3 int majorityElement(int* nums, int n) {
4     int count = 0, candidate = 0;
5
6     for (int i = 0; i < n; i++) {
7         if (count == 0) {
8             candidate = nums[i];
9             count = 1;
10        }
11        else if (nums[i] == candidate) {
12            count++;
13        }
14        else {
15            count--;
16        }
17    }
18
19    return candidate;
20}
21
22 int main() {
23     int n;
24     scanf("%d", &n);
25     int num[n];
26
27     for (int i = 0; i < n; i++) {
28         scanf("%d", &num[i]);
29     }
30
31     int result = majorityElement(num, n);
32     printf("%d\n", result);
33
34     return 0;
35 }
```

	Input	Expected	Got	
✓	2 2 2 2	2	2	✓

Passed all tests! ✓

Correct

Marked for this submission: 1.00/1.00.

Problem Statement:

Given a sorted array and a value x, the floor of x is the largest element in array smaller than or equal to x. Write divide and conquer algorithm to find floor of x.

Input Format:

First Line Contains Integer n - Size of array

Next n lines Contains n numbers - Elements of an array

Last Line Contains Integer x - Value for x

Output Format:

First Line Contains Integer - Floor value for x

Answer: (sparely regime: 0.9)

```
1  #include <stdio.h>
2  int find_floor(int arr[], int low, int high, int x, int floor) {
3      if (low > high) {
4          return floor;
5      }
6
7      int mid = (low + high) / 2;
8
9      if (arr[mid] == x) {
10         return arr[mid];
11     }
12     else if (arr[mid] > x) {
13         return find_floor(arr, low, mid - 1, x, floor);
14     }
15     else {
16         return find_floor(arr, mid + 1, high, x, arr[mid]);
17     }
18 }
19 int main() {
20     int n;
21     scanf("%d", &n);
22
23     int arr[n];
24     for (int i = 0; i < n; i++) {
25         scanf("%d", &arr[i]);
26     }
27
28     int x;
29     scanf("%d", &x);
30
31     int floor_value = find_floor(arr, 0, n - 1, x, -1);
32
33     if (floor_value == -1) {
34         printf("No floor found");
35     } else {
36         printf("%d", floor_value);
37     }
38 }
39
40 return 0;
41 }
```

	Input	Expected	Got	
✓	6 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200 201 202 203 204 205 206 207 208 209 210 211 212 213 214 215 216 217 218 219 220 221 222 223 224 225 226 227 228 229 230 231 232 233 234 235 236 237 238 239 240 241 242 243 244 245 246 247 248 249 250 251 252 253 254 255 256 257 258 259 260 261 262 263 264 265 266 267 268 269 270 271 272 273 274 275 276 277 278 279 280 281 282 283 284 285 286 287 288 289 290 291 292 293 294 295 296 297 298 299 300 301 302 303 304 305 306 307 308 309 310 311 312 313 314 315 316 317 318 319 320 321 322 323 324 325 326 327 328 329 330 331 332 333 334 335 336 337 338 339 340 341 342 343 344 345 346 347 348 349 350 351 352 353 354 355 356 357 358 359 360 361 362 363 364 365 366 367 368 369 370 371 372 373 374 375 376 377 378 379 380 381 382 383 384 385 386 387 388 389 390 391 392 393 394 395 396 397 398 399 400 401 402 403 404 405 406 407 408 409 410 411 412 413 414 415 416 417 418 419 420 421 422 423 424 425 426 427 428 429 430 431 432 433 434 435 436 437 438 439 440 441 442 443 444 445 446 447 448 449 450 451 452 453 454 455 456 457 458 459 460 461 462 463 464 465 466 467 468 469 470 471 472 473 474 475 476 477 478 479 480 481 482 483 484 485 486 487 488 489 490 491 492 493 494 495 496 497 498 499 500 501 502 503 504 505 506 507 508 509 510 511 512 513 514 515 516 517 518 519 520 521 522 523 524 525 526 527 528 529 530 531 532 533 534 535 536 537 538 539 540 541 542 543 544 545 546 547 548 549 550 551 552 553 554 555 556 557 558 559 559 560 561 562 563 564 565 566 567 568 569 569 570 571 572 573 574 575 576 577 578 579 579 580 581 582 583 584 585 586 587 588 589 589 590 591 592 593 594 595 596 597 598 599 599 600 601 602 603 604 605 606 607 608 609 609 610 611 612 613 614 615 616 617 618 619 619 620 621 622 623 624 625 626 627 628 629 629 630 631 632 633 634 635 636 637 638 639 639 640 641 642 643 644 645 646 647 648 649 649 650 651 652 653 654 655 656 657 658 659 659 660 661 662 663 664 665 666 667 668 669 669 670 671 672 673 674 675 676 677 678 679 679 680 681 682 683 684 685 686 687 688 689 689 690 691 692 693 694 695 696 697 698 699 699 700 701 702 703 704 705 706 707 708 709 709 710 711 712 713 714 715 716 717 718 719 719 720 721 722 723 724 725 726 727 728 729 729 730 731 732 733 734 735 736 737 738 739 739 740 741 742 743 744 745 746 747 748 749 749 750 751 752 753 754 755 756 757 758 759 759 760 761 762 763 764 765 766 767 768 769 769 770 771 772 773 774 775 776 777 778 779 779 780 781 782 783 784 785 786 787 788 789 789 790 791 792 793 794 795 796 797 798 799 799 800 801 802 803 804 805 806 807 808 809 809 810 811 812 813 814 815 816 817 818 819 819 820 821 822 823 824 825 826 827 828 829 829 830 831 832 833 834 835 836 837 838 839 839 840 841 842 843 844 845 846 847 848 849 849 850 851 852 853 854 855 856 857 858 859 859 860 861 862 863 864 865 866 867 868 869 869 870 871 872 873 874 875 876 877 878 879 879 880 881 882 883 884 885 886 887 888 889 889 890 891 892 893 894 895 896 897 898 899 899 900 901 902 903 904 905 906 907 908 909 909 910 911 912 913 914 915 916 917 918 919 919 920 921 922 923 924 925 926 927 928 929 929 930 931 932 933 934 935 936 937 938 939 939 940 941 942 943 944 945 946 947 948 949 949 950 951 952 953 954 955 956 957 958 959 959 960 961 962 963 964 965 966 967 968 969 969 970 971 972 973 974 975 976 977 978 979 979 980 981 982 983 984 985 986 987 988 989 989 990 991 992 993 994 995 996 997 997 998 999 999 1000 1001 1002 1003 1004 1005 1006 1007 1008 1009 1009 1010 1011 1012 1013 1014 1015 1016 1017 1018 1019 1019 1020 1021 1022 1023 1024 1025 1026 1027 1028 1029 1029 1030 1031 1032 1033 1034 1035 1036 1037 1038 1039 1039 1040 1041 1042 1043 1044 1045 1046 1047 1048 1049 1049 1050 1051 1052 1053 1054 1055 1056 1057 1058 1059 1059 1060 1061 1062 1063 1064 1065 1066 1067 1068 1069 1069 1070 1071 1072 1073 1074 1075 1076 1077 1078 1079 1079 1080 1081 1082 1083 1084 1085 1086 1087 1088 1089 1089 1090 1091 1092 1093 1094 1095 1096 1097 1097 1098 1099 1099 1100 1101 1102 1103 1104 1105 1106 1107 1108 1109 1109 1110 1111 1112 1113 1114 1115 1116 1117 1118 1119 1119 1120 1121 1122 1123 1124 1125 1126 1127 1128 1129 1129 1130 1131 1132 1133 1134 1135 1136 1137 1138 1139 1139 1140 1141 1142 1143 1144 1145 1146 1147 1148 1149 1149 1150 1151 1152 1153 1154 1155 1156 1157 1158 1159 1159 1160 1161 1162 1163 1164 1165 1166 1167 1168 1169 1169 1170 1171 1172 1173 1174 1175 1176 1177 1178 1179 1179 1180 1181 1182 1183 1184 1185 1186 1187 1188 1189 1189 1190 1191 1192 1193 1194 1195 1196 1197 1197 1198 1199 1199 1200 1201 1202 1203 1204 1205 1206 1207 1208 1209 1209 1210 1211 1212 1213 1214 1215 1216 1217 1218 1219 1219 1220 1221 1222 1223 1224 1225 1226 1227 1228 1229 1229 1230 1231 1232 1233 1234 1235 1236 1237 1238 1239 1239 1240 1241 1242 1243 1244 1245 1246 1247 1248 1249 1249 1250 1251 1252 1253 1254 1255 1256 1257 1258 1259 1259 1260 1261 1262 1263 1264 1265 1266 1267 1268 1269 1269 1270 1271 1272 1273 1274 1275 1276 1277 1278 1279 1279 1280 1281 1282 1283 1284 1285 1286 1287 1288 1289 1289 1290 1291 1292 1293 1294 1295 1296 1297 1297 1298 1299 1299 1300 1301 1302 1303 1304 1305 1306 1307 1308 1309 1309 1310 1311 1312 1313 1314 1315 1316 1317 1318 1319 1319 1320 1321 1322 1323 1324 1325 1326 1327 1328 1329 1329 1330 1331 1332 1333 1334 1335 1336 1337 1338 1339 1339 1340 1341 1342 1343 1344 1345 1346 1347 1348 1349 1349 1350 1351 1352 1353 1354 1355 1356 1357 1358 1359 1359 1360 1361 1362 1363 1364 1365 1366 1367 1368 1369 1369 1370 1371 1372 1373 1374 1375 1376 1377 1378 1379 1379 1380 1381 1382 1383 1384 1385 1386 1387 1388 1389 1389 1390 1391 1392 1393 1394 1395 1396 1397 1397 1398 1399 1399 1400 1401 1402 1403 1404 1405 1406 1407 1408 1409 1409 1410 1411 1412 1413 1414 1415 1416 1417 1418 1419 1419 1420 1421 1422 1423 1424 1425 1426 1427 1428 1429 1429 1430 1431 1432 1433 1434 1435 1436 1437 1438 1439 1439 1440 1441 1442 1443 1444 1445 1446 1447 1448 1449 1449 1450 1451 1452 1453 1454 1455 1456 1457 1458 1459 1459 1460 1461 1462 1463 1464 1465 1466 1467 1468 1469 1469 1470 1471 1472 1473 1474 1475 1476 1477 1478 1479 1479 1480 1481 1482 1483 1484 1485 1486 1487 1488 1489 1489 1490 1491 1492 1493 1494 1495 1496 1497 1497 1498 1499 1499 1500 1501 1502 1503 1504 1505 1506 1507 1508 1509 1509 1510 1511 1512 1513 1514 1515 1516 1517 1518 1519 1519 1520 1521 1522 1523 1524 1525 1526 1527 1528 1529 1529 1530 1531 1532 1533 1534 1535 1536 1537 1538 1539 1539 1540 1541 1542 1543 1544 1545 1546 1547 1548 1549 1549 1550 1551 1552 1553 1554 1555 1556 1557 1558 1559 1559 1560 1561 1562 1563 1564 1565 1566 1567 1568 1569 1569 1570 1571 1572 1573 1574 1575 1576 1577 1578 1579 1579 1580 1581 1582 1583 1584 1585 1586 1587 1588 1589 1589 1590 1591 1592 1593 1594 1595 1596 1597 1597 1598 1599 1599 1600 1601 1602 1603 1604 1605 1606 1607 1608 1609 1609 1610 1611 1612 1613 1614 1615 1616 1617 1618 1619 1619 1620 1621 1622 1623 1624 1625 1626 1627 1628 1629 1629 1630 1631 1632 1633 1634 1635 1636 1637 1638 1639 1639 1640 1641 1642 1643 1644 1645 1646 1647 1648 1649 1649 1650 1651 1652 1653 1654 1655 1656 1657 1658 1659 1659 1660 1661 1662 1663 1664 1665 1666 1667 1668 1669 1669 1670 1671 1672 1673 1674 1675 1676 1677 1678 1679 1679 1680 1681 1682 1683 1684 1685 1686 1687 1688 1689 1689 1690 1691 1692 1693 1694 1695 1696 1697 1697 1698 1699 1699 1700 1701 1702 1703 1704 1705 1706 1707 1708 1709 1709 1710 1711 1712 1713 1714 1715 1716 1717 1718 1719 1719 1720 1721 1722 1723 1724 1725 1726 1727 1728 1729 1729 1730 1731 1732 1733 1734 1735 1736 1737 1738 1739 1739 1740 1741 1742 1743 1744 1745 1746 1747 1748 1749 1749 1750 1751 1752 1753 1754 1755 1756 1757 175			

Problem Statement:

Given a sorted array of integers say arr[] and a number x. Write a recursive program using divide and conquer strategy to check if there exist two elements in the array whose sum = x. If there exist such two elements then return the integers, otherwise print as "Not".

Note: Write a Divide and Conquer Solution.

Input Format:

First Line Contains Integer n - Size of array

Next n lines Contains n numbers - Elements of an array

Last Line Contains Integer x - Sum Value

Output Format:

First Line Contains Integer - Element1

Second Line Contains Integer - Element2 (Element 1 and Element 2 together sums to value "x")

Answer: (passed regime: 0 %)

```
1 #include <stdio.h>
2
3 void find_pair(int arr[], int low, int high, int x) {
4     if (low >= high) {
5         printf("Not\n");
6         return;
7     }
8
9     int sum = arr[low] + arr[high];
10
11    if (sum == x) {
12        printf("Sum%d\n", arr[low], arr[high]);
13        return;
14    }
15    else if (sum < x) {
16        find_pair(arr, low + 1, high, x);
17    }
18    else {
19        find_pair(arr, low, high - 1, x);
20    }
21 }
22
23 int main() {
24     int n;
25     scanf("%d", &n);
26
27     int arr[n];
28     for (int i = 0; i < n; i++) {
29         scanf("%d", &arr[i]);
30     }
31
32     int x;
33     scanf("%d", &x);
34
35     find_pair(arr, 0, n - 1, x);
36
37     return 0;
38 }
```

	Input	Expected	Got	
✓	4	4	4	✓
x	10	10	10	
4				
10				
14				
16				
18				
20				
24				
26				
28				
30				
32				
34				
36				
38				
40				
42				
44				
46				
48				
50				
52				
54				
56				
58				
60				
62				
64				
66				
68				
70				
72				
74				
76				
78				
80				
82				
84				
86				
88				
90				
92				
94				
96				
98				
100				

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00

Write a Program to implement the Quick Sort Algorithm

Input Format:

The first line contains the no of elements in the list-n.
The next n lines contain the elements.

Output:

Sorted list of elements

For example:

Input	Result
5 12 45 67 78 99 87 34 52 98 76	12 34 45 67 78 99

Answer:

```
1 #include <cs50.h>
2 void swap(int *a, int *b) {
3     int temp = *a;
4     *a = *b;
5     *b = temp;
6 }
7 int partition(int arr[], int low, int high) {
8     int pivot = arr[high];
9     int i = low - 1;
10    for (int j = low; j < high; j++) {
11        if (arr[j] < pivot) {
12            i++;
13            swap(arr[i], arr[j]);
14        }
15    }
16    swap(arr[i + 1], arr[high]);
17    return i + 1;
18 }
19 void quickSort(int arr[], int low, int high) {
20    if (low < high) {
21        int pi = partition(arr, low, high);
22        quickSort(arr, low, pi - 1);
23        quickSort(arr, pi + 1, high);
24    }
25 }
26
27 int main() {
28     int n;
29     scanf("%d", &n);
30     int arr[n];
31     for (int i = 0; i < n; i++) {
32         scanf("%d", &arr[i]);
33     }
34     quickSort(arr, 0, n - 1);
35     for (int i = 0; i < n; i++) {
36         printf("%d ", arr[i]);
37     }
38     printf("\n");
39     return 0;
40 }
```

	Input	Expected	Got	
✓	5 12 45 67 78 99	12 34 45 67 78 99	12 45 67 78 99	✓
✓	10 1 56 78 90 32 56 15 38 90 111	1 56 15 32 56 78 90 90 111	1 56 15 32 56 78 90 90 111	✓
✓	12 9 8 7 6 5 4 3 2 1 10 11 99	1 2 3 4 5 6 7 8 9 10 11 99	1 2 3 4 5 6 7 8 9 10 11 99	✓

Passed all tests! ✓

Correct

Stars for this submission: 1.00/1.00

Write a program to take value V and we want to make change for V Rs, and we have infinite supply of each of the denominations in Indian currency, i.e., we have infinite supply of {1, 2, 5, 10, 20, 50, 100, 500, 1000} valued coins/notes, what is the minimum number of coins and/or notes needed to make the change.

Input Format:

Take an integer from stdin.

Output Format:

print the integer which is change of the number.

Example Input :

64

Output:

4

Explanation:

We need a 50 Rs note and a 10 Rs note and two 2 rupee coins.

Answer: (penalty regime: 0 %)

```
1 //include <stdio.h>
2
3 int main() {
4     int V;
5     scanf("%d", &V);
6
7     int denominations[] = {1000, 500, 100, 50, 20, 10, 5, 2, 1};
8     int count = 0;
9
10    for (int i = 0; i < 9; i++) {
11        while (V >= denominations[i]) {
12            V -= denominations[i];
13            count++;
14        }
15    }
16
17    printf("%d\n", count);
18    return 0;
19 }
```

	Input	Expected	Got	
✓	64	64	64	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Final score:

Answer you are an infinite parent and want to give your children some cookies, that you should give each child at least one cookie.

Each child has a greed factor $g[i]$, which is the minimum size of a cookie that the child will be content with, and each cookie has a size $s[j]$. If $s[j] \geq g[i]$, we can assign the cookie j to the child i , and the child will be content. Your goal is to minimize the number of your unused cookies and output the maximum number.

Example 0:

Input:

```
1
2 3 5
3
1 1
```

Output:

```
1
```

Explanation: You have 3 children and 5 cookies. The greed factors of 3 children are 1, 1, 3.

And even though you have 5 cookies, since their sizes are both 1, you could only make the children's greed factors 1 content.

So you need to output 1.

Constraints:

- 1 <= $g.length \leq 5000$
- 1 <= $s.length \leq 5000$
- 1 <= $g[i], s[j] \leq 2000$

Answer: (possibly requires 0ms)

```
1 // Please do not modify or delete this line.
2 // It is required by our code checker.
3
4 class Solution {
5     public int findContentChildren(int[] g, int[] s) {
6         int ans = 0;
7         for (int i = 0; i < g.length; i++) {
8             for (int j = 0; j < s.length; j++) {
9                 if (s[j] >= g[i]) {
10                     ans++;
11                     s[j] = -1;
12                     break;
13                 }
14             }
15         }
16         return ans;
17     }
18 }
```

	Input	Expected	Got
#1	1 2 3 5 3 1 1	1	1
#2			
#3			
#4			

Passed all tests (#1)

Correct

Work in progress submission (7001 ms)

A person needs to eat burgers. Each burger contains a dozen of calories. After eating the burger, the person needs to run a distance to burn out his calories. If he has eaten i burgers with c calories each, then he has to run at least $i^2 \times c$ kilometers to burn out the calories. For example, if he ate 4 burgers with the count of calories in the order: $[1, 2, 2]$, the kilometers he needs to run are $(1^2 \times 1) + (2^2 \times 2) + (2^2 \times 2) = 1 + 8 + 16 = 25$. But this is not the answer, we need to try out other orders of consumption and choose the minimum value. Determine the minimum distance he needs to run. Note: We can eat burger in any order and use an efficient sorting algorithm. Apply greedy approach to solve the problem.

Input Format

First line contains the number of burgers

Second line contains calories of each burger which is a space-separated integers

Output Format

Print: Minimum number of kilometers needed to run to burn out the calories

Sample Input

```
4
1 16 2
```

Sample Output

```
25
```

For example:

Test	Input	Result
Test Case 1	4 1 16 2	25

Answer: (penalty regimen: 0 %)

```
1 //include <stdio.h>
2 //include <math.h>
3
4
5 int cmp(const void *a,const void *b)
6 {
7     return((int*)b)-(int*)a;
8 }
9
10 int main(){
11     int n;
12     scanf("%d",&n);
13     int a[n];
14     for(int i=0;i<n;i++)
15         scanf("%d",&a[i]);
16     qsort(a,n,sizeof(int),cmp);
17     int min=0;
18     for(int i=0;i<n;i++)
19         min+=i*a[i];
20     printf("%d\n",min);
21 }
```

	Test	Input	Expected	Got	
✓	Test Case 1	4 1 16 2	25	25	✓
✓	Test Case 2	5 2 1 8 6	100	100	✓
✓	Test Case 3	3 3 16 7	76	76	✓

Passed all testcases

Correct

Marks for this submission: 1.00/1.00.

Given an array of N integers, we have to maximize the sum of $arr[i] * i$, where i is the index of the element ($i = 0, 1, 2, \dots, N$). Write an algorithm based on Greedy technique with a Complexity $O(n\log n)$.

Input Format:

The first line specifies the number of elements - n .

The next n lines contain the array elements.

Output Format:

Maximum Array Sum to be printed.

Sample Input:

5

2 5 3 4 0

Sample Output:

40

Answer: (Complexity: $O(n\log n)$)

```
1 #include <stdio.h>
2
3 int main() {
4     int n, sum = 0;
5     scanf("%d", &n);
6     int arr[n];
7     for (int i = 0; i < n; i++) {
8         scanf("%d", &arr[i]);
9     }
10    for (int i = 0; i < n - 1; i++) {
11        for (int j = i + 1; j < n - i - 1; j++) {
12            if (arr[i] > arr[j + 1]) {
13                int temp = arr[i];
14                arr[i] = arr[j + 1];
15                arr[j + 1] = temp;
16            }
17        }
18    }
19    for (int i = 0; i < n; i++) {
20        sum += arr[i] * i;
21    }
22    printf("%d", sum);
23 }
```

	Input	Expected	Got	
✓	5 2 5 3 4 0	40	40	✓
✓	10 2 2 2 1 6 3 4 5	190	190	✓
✓	2 6 5 2	54	54	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Given two arrays `array_1[]` and `array_2[]` of same size `N`. We need to first rearrange the arrays such that the sum of the product of pair `i` element from each() is minimum, that is $\text{SUM}(A[i] * B[i])$ for all i is minimum.

For example:

Input	Result
<code>x</code>	<code>20</code>
<code>y</code>	
<code>z</code>	
<code>t</code>	
<code>s</code>	
<code>v</code>	
<code>u</code>	

Answer: (penalty regime: 0/8)

```
1 #include<iostream.h>
2 #include<cmath.h>
3 int compare(const void *a, const void *b) {
4     return *(int *)a - *(int *)b;
5 }
6 int comparedesc(const void *a, const void *b) {
7     return *(int *)b - *(int *)a;
8 }
9 long minProductSum(int arr1[], int arr2[], int n) {
10    quicksort(arr1, n, start(0), compare);
11    quicksort(arr2, n, start(0), comparedesc);
12    long sum = 0;
13    for (int i = 0; i < n; i++) {
14        sum += arr1[i] * arr2[i];
15    }
16    return sum;
17 }
18 int main() {
19     int n;
20     scanf("%d", &n);
21     int arr1[n], arr2[n];
22     for (int i = 0; i < n; i++) {
23         scanf("%d", &arr1[i]);
24     }
25     for (int i = 0; i < n; i++) {
26         scanf("%d", &arr2[i]);
27     }
28     long n = sizeof(arr1) / sizeof(arr1[0]);
29     long result = minProductSum(arr1, arr2, n);
30     printf("%d", result);
31 }
32 }
```

	Input	Expected	Got	
✓	<code>x</code> <code>y</code> <code>z</code> <code>t</code> <code>s</code> <code>v</code> <code>u</code>	<code>20</code>	<code>24</code>	✓
✓	<code>x</code> <code>y</code> <code>z</code> <code>t</code> <code>s</code> <code>v</code> <code>u</code> <code>w</code>	<code>20</code>	<code>22</code>	✓
✓	<code>x</code> <code>20</code> <code>30</code> <code>40</code> <code>50</code> <code>60</code> <code>70</code> <code>80</code>	<code>100</code>	<code>100</code>	✓

Passed all tests! ✓

Correct

Score for this submission: 1.00/1.00.

Playing with Numbers:

Ram and Sita are playing with numbers by giving puzzles to each other. Now it was Ram's turn, so he gave Sita a positive integer N and two numbers 1 and 3. He asked her to find the possible ways by which the number n can be represented using 1 and 3. Write any efficient algorithm to find the possible ways.

Example 1:

Input: 6

Output:

Explanation: There are 8 ways to represent number with 1 and 3.

$1+1+1+1+1+1$

$3+3$

$1+1+1+2$

$1+1+3$

$1+3+1$

$3+1+1$

$2+1+1+1$

Input Format:

First Line contains the number n .

Output Format:

Print: The number of possible ways ' n ' can be represented using 1 and 3.

Sample Input:

6

Sample Output:

8

Answer: (penalty regime: 0 %)

```
1 //include <iostream.h>
2
3 int main() {
4     int n;
5     scanf("%d", &n);
6
7     long long dp[n+1];
8     for (int i = 0; i <= n; i++) {
9         dp[i] = 0;
10    }
11
12    dp[0] = 1;
13
14    for (int i = 1; i <= n; i++) {
15        dp[i] += dp[i - 1];
16        if (i >= 2) {
17            dp[i] += dp[i - 2];
18        }
19    }
20
21    printf("%d\n", dp[n]);
22    return 0;
23 }
```

	Input	Expected	Got	
✓	6	8	8	✓
✓	24	1677	1677	✓
✓	100	2108201999721628	2108201999721628	✓

Passed all testcases!

Correct

Starts from this submission: 10.06/10.00.

Playing with Chessboard:

Item is given with an $n \times n$ chessboard with each cell with a monetary value. Item stands at the (0,0), that is the position of the top-left white rook. He is given a task to reach the bottom-right black rook position ($(n-1, n-1)$) constrained that he needs to reach the position by traveling the maximum monetary path under the condition that he can only travel one step right or one step down the board. Help him to achieve it by providing an efficient DP algorithm.

Example:**Input:**

3

1 2 4

2 3 4

8 7 1

Output:

19

Explanation:

Total there will be 6 paths, among that the optimal is:

Optimal path value: $1+2+8+7+1 = 19$

Input Format

First Line contains the integer n .

The next n lines contain the $n \times n$ chessboard values.

Output Format

Print Maximum monetary value of the path.

Answer: (penalty: negative 0.0%)

```
1 //function optimise
2
3 int max(int a, int b) {
4     return a > b ? a : b;
5 }
6
7 int min() {
8     cout<<"Min";
9 }
10
11 int board[n][n];
12 int dp[n][n];
13
14 for (int i = 0; i < n; i++) {
15     for (int j = 0; j < n; j++) {
16         cout<<"<*>"<< board[i][j];
17     }
18 }
19
20 dp[0][0] = board[0][0];
21
22 for (int i = 1; i < n; i++) {
23     dp[0][i] = dp[0][i - 1] + board[0][i];
24 }
25
26 for (int j = 1; j < n; j++) {
27     dp[j][0] = dp[j - 1][0] + board[j][0];
28 }
29
30 for (int i = 1; i < n; i++) {
31     for (int j = 1; j < n; j++) {
32         dp[i][j] = max(dp[i - 1][j], dp[i][j - 1]) + board[i][j];
33     }
34 }
35
36 printf("Max<*> %d<*> dp[%d][%d]\n", n - 1);
37
38 }
39 }
```

Input	Expected	Got
3 1 2 4 2 3 4 8 7 1	19	19 ✓
4 1 2 3 4 2 3 4 5 3 4 5 6 4 5 6 7	28	28 ✓
4 1 2 3 4 2 3 4 5 3 4 5 6 4 5 6 7	28	28 ✓

Passed all tests! ✓

Correct

Starts for this submission: 11.09.10.00

Given two strings find the length of the common longest subsequence (need not be contiguous) between the two.

Example:

s1: gatadb

s2: gatedab

s1	a	g	t	a	b		
s2	g	x	t	x	a	y	b

The length is 4

Solving it using Dynamic Programming

For example:

Input	Result
abk	2
abk	

Answer: (possibly regions: Q %)

```
1 #include <string.h>
2 #include <string.h>
3
4 int max(int a, int b) {
5     return a > b ? a : b;
6 }
7
8 int main() {
9     char s1[100], s2[100];
10    scanf("%s", s1);
11    scanf("%s", s2);
12
13    int n = strlen(s1);
14    int m = strlen(s2);
15
16    int dp[n + 1][m + 1];
17
18    for (int i = 0; i < n; i++) {
19        for (int j = 0; j < m; j++) {
20            if (i == 0 || j == 0)
21                dp[i][j] = 0;
22            else if (s1[i - 1] == s2[j - 1])
23                dp[i][j] = dp[i - 1][j - 1] + 1;
24            else
25                dp[i][j] = max(dp[i - 1][j], dp[i][j - 1]);
26        }
27    }
28
29    printf("%d\n", dp[n][m]);
30
31    return 0;
32 }
```

	Input	expected	Got	
✓	abk	2	2	✓
✓	abk			

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Problem statement:

Find the length of the Longest Non-decreasing Subsequence in a given Sequence.

Ig:

Input:9

Sequence:[-1,3,4,5,2,2,2,2,3]

The subsequence is [-1,2,2,2,2,3]

Output:6

Answer: (penalty regime: 0 %)

```
1 //include <iostream.h>
2
3 int max(int a, int b) {
4     return a > b ? a : b;
5 }
6
7 int main() {
8     int n;
9     cin>>n;
10    int A[n];
11    for (int i = 0; i < n; i++) {
12        cin>>A[i];
13    }
14
15    int dp[n];
16    for (int i = 0; i < n; i++) {
17        dp[i] = 1;
18    }
19
20    for (int i = 1; i < n; i++) {
21        for (int j = 0; j < i; j++) {
22            if (A[i] >= A[j]) {
23                dp[i] = max(dp[i], dp[j] + 1);
24            }
25        }
26    }
27
28    int maxlen = 0;
29    for (int i = 0; i < n; i++) {
30        if (dp[i] > maxlen) {
31            maxlen = dp[i];
32        }
33    }
34
35    printf("%d\n", maxlen);
36    return 0;
37 }
38 }
```

	Input	Expected	Got	
✓	9 -1,3,4,5,2,2,2,2,3	9	9	✓
✓	7 1,2,2,4,5,7,6	6	6	✓

Passed all tests! ✓

Correct

Mark for this submission: 1.00/1.00.

Find Duplicate in Array.

Given a read-only array of n integers between 1 and n , find one number that repeats.

Input Format:

First Line - Number of elements

n Lines - n elements

Output Format:

Element x - That is repeated

For example:

Input	Result
5	1
1 1 2 3 5	

Answer: (passed) regime: 0.9%

```
1 //include <stdio.h>
2
3 int main() {
4     int n;
5     scanf("%d", &n);
6
7     int A[n + 1];
8     for (int i = 0; i < n; i++) {
9         scanf("%d", &A[i]);
10    }
11
12    int freq[n + 1];
13    for (int i = 0; i < n; i++) {
14        freq[i] = 0;
15    }
16
17    for (int i = 0; i < n; i++) {
18        if (freq[A[i]] == 1) {
19            printf("Min: %d\n", A[i]);
20            return 0;
21        }
22        freq[A[i]] += 1;
23    }
24
25    return 0;
26}
27
```

	Input	Expected	Got	
✓	11 1 8 9 7 6 5 1 2 3 4 5 1	?	2	✓
✓	5 1 2 3 4 5	4	4	✓
✓	5 1 1 2 3 4 5	1	1	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Post the intersection of two sorted arrays.

Given 2 sorted arrays, find all the elements which occur in both the arrays.

Input Format:

- The first line contains, T , the number of test cases. Following T lines contain:
- 1. Case 1 contains M_1 , followed by M_1 integers of the first array.
- 2. Case 2 contains M_2 , followed by M_2 integers of the second array.

Output Format:

The intersection of the arrays, in a single line.

Example:

Input:

```
1
2 3 4 5 6
4 5 7 8 9 10
```

Output:

```
5 6
```

Input:

```
2
1 2 3 4 5
2 3 4
```

Output:

```
3 4
```

For example:

Input	Result
1	10 11
2 3 4 5 6	
4 5 7 8 9 10	

Answers: (possibly right or wrong)

```
1 function minID {
2     let minID = 1;
3     for (let i = 0; i < arr1.length; i++) {
4         if (arr1[i] < minID) {
5             minID = arr1[i];
6         }
7     }
8     return minID;
9 }
10
11 function maxID {
12     let maxID = 1;
13     for (let i = 0; i < arr2.length; i++) {
14         if (arr2[i] > maxID) {
15             maxID = arr2[i];
16         }
17     }
18     return maxID;
19 }
20
21 function intersection(arr1, arr2) {
22     let result = [];
23     let minID = minID(arr1);
24     let maxID = maxID(arr2);
25     for (let i = minID; i <= maxID; i++) {
26         if (arr1.indexOf(i) > -1 && arr2.indexOf(i) > -1) {
27             result.push(i);
28         }
29     }
30     return result;
31 }
```

Input	Expected	Get	Diff
1 2 3 4 5 6 4 5 7 8 9 10	5 6	5 6	✓
2 1 2 3 4 5 2 3 4	3 4	3 4	✓

Possible result: ✓

Success

Score for this submission: 200.00

Find the intersection of two sorted arrays.

Given two sorted arrays, find all the elements which occur in both the arrays.

Input Format:

- The first line contains T , the number of test cases, followed by T lines containing:
- Line 1 contains M , followed by M integers of the first array.
- Line 2 contains N , followed by N integers of the second array.

Output Format:

The intersection of the arrays is a single line.

Example:

Input:

```
1
5
1 2 3 4 5
2 2 3 3 4 5 6 7 8 9
```

Output:

```
2 3 4
```

Input:

```
2
5
1 2 3 4 5
2 2 3 3 4 5 6 7 8 9
5
1 2 3 4 5
2 2 3 3 4 5 6 7 8 9
```

Output:

```
2 3 4
```

For example:

Input	Result
1 5 1 2 3 4 5	1 2 3 4 5
2 5 1 2 3 4 5 5 1 2 3 4 5	1 2 3 4
2 3 4 5 6 7 8 9 2 3 4 5 6 7 8 9	2 3 4

Answers: (using engine: D9)

```
2
public class Solution {
    3     public static void main(String[] args) {
    4         int m;
    5         int n;
    6         int[] arr1 = {1, 2, 3, 4, 5};
    7         int[] arr2 = {2, 2, 3, 3, 4, 5, 6, 7, 8, 9};
    8         m = arr1.length;
    9         n = arr2.length;
    10        int i = 0;
    11        int j = 0;
    12        int[] result = new int[m + n];
    13        int k = 0;
    14        while (i < m && j < n) {
    15            if (arr1[i] == arr2[j]) {
    16                result[k] = arr1[i];
    17                k++;
    18                i++;
    19                j++;
    20            } else if (arr1[i] < arr2[j]) {
    21                i++;
    22            } else {
    23                j++;
    24            }
    25        }
    26        if (i < m) {
    27            for (int l = i; l < m; l++) {
    28                result[k] = arr1[l];
    29                k++;
    30            }
    31        }
    32        if (j < n) {
    33            for (int l = j; l < n; l++) {
    34                result[k] = arr2[l];
    35                k++;
    36            }
    37        }
    38        System.out.println(Arrays.toString(result));
    39    }
}
```

Input	Expected	Our	Diff
1 5 1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	✓
2 5 1 2 3 4 5 5 1 2 3 4 5	1 2 3 4	1 2 3 4	✓
2 3 4 5 6 7 8 9 2 3 4 5 6 7 8 9	2 3 4	2 3 4	✓

Percent of cases: 100

Details:

Score for this submission: 1.00/1.00

Given an array A of sorted integers and another non-negative integer k, find if there exists 2 indices i and j such that $A[i] \cdot A[j] = k$ ($i \neq j$).

Input Format:

First Line n - Number of elements in an array

Next n Lines - N elements in the array

k - Non - Negative Integer

Output Format:

1 - If pair exists

0 - If no pair exists

Explanation for the given Sample Testcase:

YES as $5 \times 1 = 4$

So return 1.

For example:

Input	Result
5	1
5 1 5	
5	

Answer: (penalty regime: 0 %)

```
1 #include <stdio.h>
2
3 int main() {
4     int n, k;
5     scanf("%d", &n);
6
7     int A[n];
8     for (int i = 0; i < n; i++) {
9         scanf("%d", &A[i]);
10    }
11
12    scanf("%d", &k);
13
14    int i = 0, j = 1;
15    while (i < j && i < n) {
16        int diff = A[i] * A[j];
17        if (diff == k || k % diff == 0) {
18            printf("%d\n");
19            return 1;
20        } else if (diff < k) {
21            j++;
22        } else {
23            i++;
24            if (i == j) {
25                j++;
26            }
27        }
28    }
29
30    printf("No\n");
31    return 0;
32 }
```

	Input	Expected	Got	
✓	5 5 1 5 5	1	1	✓
✓	10 1 5 6 8 12 21 35 26 23 26 1	1	1	✓
✓	10 1 2 3 5 11 51 16 21 28 29 0	0	0	✓
✓	10 0 2 3 7 13 31 33 39 41 26 10	1	1	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00

Given an array A of sorted integers and another non-negative integer k, find if there exists 2 indices i and j such that $A[i] \cdot A[j] = k$, $i < j$.

Input Format:

First Line: n - Number of elements in an array.

Next n Lines - N elements in the array.

k - Non - Negative Integer

Output Format:

1 - If pair exists

0 - If no pair exists

Explanation for the given Sample Testcase:

YES as $5 \times 4 = 20$

So Return 1.

For example:

Input	Result
5	1
1 2 5	
4	

Answer: (penalty regime: 0 %)

```
1 //Complete catalin.ho
2
3 int main() {
4     int n, k;
5     scanf("%d", &n);
6     int A[n];
7     for (int i = 0; i < n; i++) {
8         scanf("%d", &A[i]);
9     }
10
11    scanf("%d", &k);
12
13    int i = 0, j = 1;
14    while (i < n && j < n) {
15        int diff = A[i] * A[j];
16        if (diff == k) {
17            printf("1\n");
18            return 0;
19        } else if (diff < k) {
20            i++;
21        } else {
22            j++;
23        }
24    }
25
26    printf("0\n");
27    return 0;
28}
```

	Input	Expected	Got	
✓	5 1 2 5 5	1	1	✓
✓	10 3 5 6 8 12 14 15 18 20 25 1	1	1	✓
✓	10 1 2 3 5 12 13 15 20 25 29 0	0	0	✓
✓	10 6 2 6 7 14 19 15 28 25 26 54	1	1	✓

Passed all tests! ✓

Correct

Waited for this submission: 1.60150s