

Observações

1. As questões 1, 2 e 3 exigem vetores e repetição(ões) simples
2. As questões 4 e 5 exigem funções com passagem de vetores como parâmetros
3. As questões 6, 7 e 8 exigem vetores e repetições aninhadas (embora a 6 também possa ser resolvida com vetores e repetição simples)

Questões

1. Escreva um programa que leia um número inteiro n representando uma quantidade de meses ($1 \leq n \leq 20$). Em seguida faça a leitura de n inteiros, representando a quantidade de dias que choveu em cada um desses meses (do mês 1 até o mês n). O programa deve em seguida escrever na tela o número do mês em que mais teve dias de chuva e a respectiva quantidade. Se houver empate entre mais de um mês, escreva-os todos. Assuma que choveu em pelo menos algum dia.

Exemplos:

Input	Output
4 9 9 9 9	1 2 3 4 9
5 2 5 3 4 1	2 5

2. ▷ O MEC precisa de sua ajuda para automatizar a correção das provas objetivas do ENEM! Escreva um programa que leia um número inteiro n representando o número de questões ($1 \leq n \leq 20$). Em seguida leia as n respostas do gabarito e, em seguida, as n respostas do aluno. Assuma que as respostas estão sempre entre 1 e 5. Depois o programa deve escrever na tela quantas questões o aluno acertou e a string “acertos” ou “acerto” (para 1 acerto), conforme exemplo abaixo.

Exemplos:

Input	Output
4 1 2 3 4 1 5 3 5	2 acertos
7 1 2 3 2 1 5 4 3 3 3 3 3 3 3	1 acerto

3. Um professor de ITP quer saber quantos alunos ficaram com a nota acima da média de todas as notas (literalmente a média de todas as notas e não a média 7). Escreva um programa que leia um número inteiro n (assuma $1 \leq n \leq 30$) e, em seguida, n números reais representando as notas dos n alunos. Depois o programa deve escrever na tela quantos alunos tiveram nota acima da média (a média deve ser calculada).

Exemplo:

Input	Output
28	15
7.00 7.60 4.21 0.28 7.57 3.75 9.25 5.64	
9.75 2.93 7.02 7.04 6.31 2.37 6.19	
9.47 8.00 4.79 5.44 2.47 1.15 9.42 9.69	
5.98 5.64 9.47 8.46 2.04	

4. Crie uma função que receba dois vetores como parâmetro que representam os coeficientes (real) e os expoentes (inteiros) de um polinômio, assim como um inteiro n representando os tamanhos desses dois vetores e um número real x . A função deve retornar a avaliação do polinômio em x .

A assinatura da função deve ser a seguinte:

```
float avaliarPolinomio(float coef[], int exp[], int n, float x);
```

Por exemplo, suponha que o vetor dos coeficientes seja $[2.03, 3.84, 1.72]$, o vetor de expoentes seja $[4, 2, 0]$ e $x = 3.14$, então a função deverá retornar o valor de:

$$2.03x^4 + 3.84x^2 + 1.72x^0$$

para $x = 3.14$, o que resulta em 236.9206396848.

A função main deve ler do usuário um inteiro n (assuma $1 \leq n \leq 30$), n números reais representando os coeficientes, n números inteiros representando os expoentes e um número real x . O programa deve entrão escrever na tela, com duas casas decimais de precisão, a avaliação do polinômio em x .

Exemplo:

Input	Output
3 2.03 3.84 1.72 4 2 0 3.14	236.92

5. \triangleright Escreva a seguinte função que recebe como parâmetro um vetor v de n números reais:

```
float norma(float v[], int n)
```

A função deve retornar a norma euclidiana dada por:

$$\sqrt{v_0^2 + v_1^2 + \dots + v_{n-1}^2}$$

A função main deve ler do usuário um inteiro n (assuma $1 \leq n \leq 30$), n números reais e escrever na tela, com duas casas decimais de precisão, a norma de u fazendo uso da função norma.

6. Escreva um programa em C que leia um número inteiro n (assuma $1 \leq n \leq 10$) e, em seguida, n números inteiros. O programa deve escrever **S** se esses n números são uma permutação dos números entre 1 e n e **N** caso contrário. Por exemplo, se $n = 5$, então $[1\ 3\ 4\ 2\ 5]$ e $[5\ 2\ 3\ 1\ 4]$ são permutações de $[1\ 2\ 3\ 4\ 5]$, enquanto não são: $[1\ 6\ 4\ 3\ 2]$ (pois falta o 5) e $[1\ 2\ 5\ 3\ 3]$ (pois falta o 4).

Exemplos:

Input	Output
4 1 4 3 7	N
4 1 4 3 2	S

7. Um raio nunca cai duas vezes no mesmo lugar, é o que dizem. Você foi contratado por um instituto de pesquisas para verificar se isso é verdade ou não, ao menos com os dados disponíveis. Crie um programa que leia um número inteiro n representando a quantidade de raios registrados (assuma $n \leq 20$), seguido de n coordenadas (x,y) (em metros) representando o local onde cada raio atingiu o solo. Dois raios caem no mesmo lugar se a distância euclidiana entre as coordenadas for menor ou igual a 1 metro:

$$\sqrt{\Delta x^2 + \Delta y^2} \leq 1$$

onde Δx é a diferença entre as abscissas e Δy é a diferença entre as ordenadas.

O programa deve escrever **S** na tela caso haja dois raios que caíram no mesmo local e **N** caso contrário.

Exemplo 1 (vide imagens):

Input	Output
3	S
3.2 4.4	
3.3 5.1	
9.1 4.2	
3	N
3.2 4.4	
5.2 5.1	
9.1 4.2	

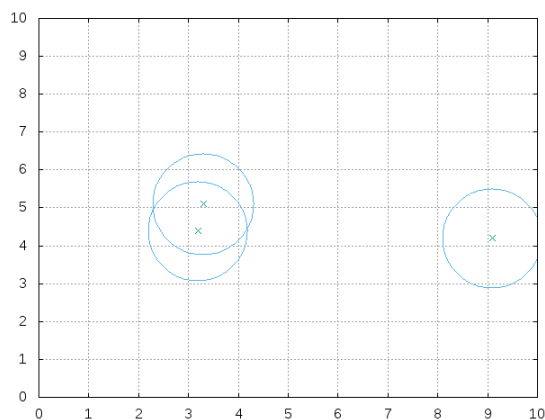


Figure 1: Localização de cada um dos raios do exemplo 1. A distância euclidiana entre o primeiro raio (3.2, 4.4) e o segundo (3.3, 5.1) é menor ou igual a 1.

8. \triangleright Escreva um programa em C que leia um número n ($n \leq 50$), leia n números inteiros $v_0 \dots v_{n-1}$, um número x e, em seguida, escreva na tela quantas são as possibilidades em que a soma de dois números v_i e v_j ($i \neq j$) dos n números digitados resulta em x .

Exemplo:

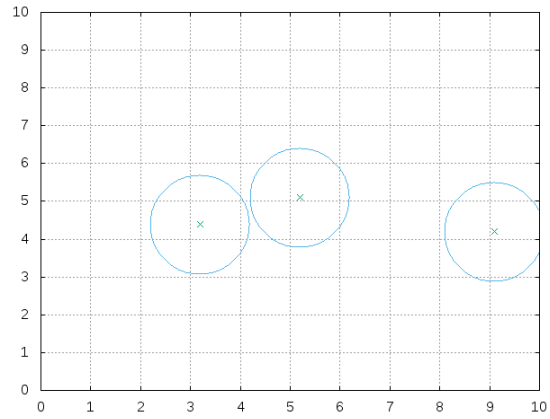


Figure 2: Localização de cada um dos raios do exemplo 2. Não há dois raios cuja distância euclidiana é menor ou igual a 1.

Input	Output
13 6 4 11 -4 12 -5 2 -1 7 -2 -3 1 8 7	4

Comentário: nesse exemplo, as possibilidades são: (6, 1), (-1, 8), (11,-4) e (12, -5).