

Prova-03

Prof. Msc. Elias Batista Ferreira
Prof. Dr. Gustavo Teodoro Laureano
Profa. Dra. Luciana Berretta
Prof. Dr. Thierson Rosa Couto

Sumário

1	Estatística Empresarial (++)	2
2	Loteria (+++)	3
3	Polinômios (++)	5

1 Estatística Empresarial (++)



(++)

Uma empresa deseja obter informações estatísticas sobre seus funcionários. Para isso, irá colher os seguintes dados dos funcionários: matrícula, idade, número de filhos, sexo e salário.

Essas informações precisam ser armazenadas em uma estrutura:

```
1 typedef struct {
2     int matricula;
3     int idade;
4     int numFilhos;
5     char sexo;
6     double salario;
7 } FUNCIONARIO;
```

Entrada

Na primeira linha ha um inteiro n , $1 < n < 500$, representando a quantidade de funcionário (fazer alocação dinâmica). A seguir haverá n linhas com n os seguintes dados separados por um espaço em branco cada: matrícula, idade, número de filhos, sexo e salário.

Saída

Deverá imprimir 4 (quatro) respostas:

- Quantidade de funcionários com idade superior a média de idades E salário superior a 3 salários mínimos.
- Quantidade de mulheres que possuem quantidade de filhos acima da média geral.
- Quantidade de homens que possuem quantidade de filhos acima da média geral.
- Quantidade de funcionários maiores de 47 anos com renda per-capita (por pessoa) abaixo de 2 salários mínimos.

* considere o salário mínimo igual a 1200.00.

Exemplo

Entrada	Saída
10 101 44 4 M 7001.00 105 56 2 F 2950.00 211 60 2 F 6870.00 221 25 1 F 9200.00 231 38 3 M 4350.00 300 70 4 M 2100.00 545 27 0 F 4500.00 654 65 1 F 2900.00 670 53 2 M 3300.00 888 55 2 F 4100.00	2 0 3 6

2 Loteria (+++)



(+++)

A Loteria é um jogo que paga um prêmio em dinheiro para o apostador que conseguir acertar os 6 números sorteados. Ainda é possível ganhar prêmios ao acertar 4 ou 5 números dentre os 60 disponíveis no volante de apostas. Para isso, você deve *marcar* 6 números do **volante**. Você poderá fazer quantas apostas quiser, ou seja, poderá jogar quantos volantes necessitar. Os números estão entre 1 e 60.

Faça um programa que receba os jogos de um apostador, em seguida, leia o resultado da loteria e verifique se o apostador acertou os números sorteados. Se o apostador acertou 4, 5 ou 6 números é necessário emitir um aviso reportando o fato.

É obrigatório utilizar estrutura para armazenar os números apostados e o resultado .

```
1 typedef struct {  
2     int numJogo;  
3     int numero[6];  
4 } CARTELA;
```

Entrada

A entrada contém vários casos de teste. A primeira linha de cada caso de teste contém um inteiro $N(1 \leq N \leq 10^3)$, indicando a quantidade de apostas do jogador. As N linhas seguintes contém o número do jogo e 6 números correspondentes aos palpites do jogador.

Em seguida, deverá ter um linha para ler o número do concurso e os 6 números sorteados, que devem ser armazenados em outra estrutura.

*** Deve-se utilizar alocação dinâmica para reservar N espaços das apostas.**

Saída

Para cada entrada, deve-se verificar se o apostador acertou, no mínimo, 4 números e emitir a seguinte mensagem:

1. QUADRA jogo: a b c d: quando a apostador acertar 4 números.
2. QUINA jogo: a b c d e: quando a apostador acertar 5 números.
3. SENA jogo: a b c d e f: quando a apostador acertar 6 números.

Após analisar todas as apostas e constatar que o apostador não conseguiu acertar, no mínimo, 4 números, escreva a mensagem "NENHUMA CARTELA PREMIADA PARA O CONCURSO concurso".

Exemplos

Entrada						
4						
1	5	15	25	35	45	55
2	9	13	28	46	51	52
3	2	28	46	47	51	13
4	8	15	25	35	45	55
1050	9	13	28	46	51	52
Saída						
SENA	2:	9	13	28	46	51 52
QUADRA	3:	28	46	51	13	

Entrada									
3									
1	3	11	44	50	56	32			
2	2	12	57	51	45	33			
3	1	34	13	46	58	52			
1051	5	15	36	47	53	60			
Saída									
NENHUMA CARTELA PREMIADA PARA O CONCURSO 1051									

3 Polinômios (++)



(++)

Faça um programa que implemente a leitura e a soma, subtração e multiplicação de uma sequência de polinômios de qualquer ordem. Neste exercício você deverá usar a estrutura `Poly`, disponível no código abaixo, para armazenar um polinômio. Nessa estrutura, o atributo `ordem` representa a maior ordem do polinômio e o vetor `coef` representa os coeficientes do polinômio. Os coeficientes são armazenados de modo que sua potência é o seu índice correspondente. Por exemplo, a representação do polinômio $2x^3 - 1x^2 + 1$ é: `ordem=3` e `coef={1,0,-1,2}`.

```
1 typedef struct {
2     int ordem;      // Ordem do polinomio
3     double * coef;  // Coeficientes. Cada indice representa a potência do coeficiente.
4 } Poly;
```

Neste exercício, a impressão de um polinômio segue o seguinte padrão: $s_c c_p x^{p_c}$, onde s_c é o sinal do coeficiente, c_p é o coeficiente da potência p e p_c é a potência do coeficiente c . Desse modo, o polinômio dado como exemplo no parágrafo anterior seria impresso como: $+2.0x^3 - 1.0x^2 + 0.0x^1 + 1.0x^0$. Note que deve ser usada somente uma casa decimal.

Você deverá implementar as funções faltantes no código abaixo.

```
1 typedef struct {
2     int ordem;      // Ordem do polinomio
3     double * coef;  // Vetor de coeficientes
4 } Poly;
5
6 /**
7  * Funcao que cria um polinomio com alocao dinamica
8  * @param Poly* Ponteiro para o novo polinomio
9  */
10 Poly * poly_new(int ord);
11
12 /**
13  * Funcao que imprime um polinomio na tela
14  * @param p Ponteiro para o polinomio
15  */
16 void poly_print(Poly * p);
17
18 /**
19  * Funcao que libera a memoria alocada a um polinomio
20  * @param p Ponteiro para o polinomio
21  */
22 void poly_free(Poly * p);
23
24 /**
25  * Cria o polinomio resultante da soma
26  * @param A Ponteiro para o primeiro polinomio
27  * @param B Ponteiro para o segundo polinomio
28  * @return Poly* <- A + B
29  */
30 Poly * poly_sum(Poly * A, Poly * B);
31
32 /**
33  * Cria o polinomio resultante da subtracao
34  * @param A Ponteiro para o primeiro polinomio
35  * @param B Ponteiro para o segundo polinomio
36  * @return Poly* <- A - B
37  */
38 Poly * poly_sub(Poly * A, Poly * B);
39
40 /**
```

```

40 * Cria o polinomio resultante da multiplicacao
41 * @param A Ponteiro para o primeiro polinomio
42 * @param B Ponteiro para o segundo polinomio
43 * @return Poly* <- A * B
44 */
45 Poly * poly_mult( Poly * A, Poly * B );
46
47 int main() {
48     Poly **P;    // Vetor de polinomios
49     int n;       // Quantidade de casos
50     // Demais declaracoes
51     // ...
52
53     scanf("%d", &n); // Definicao da quantidade de polinomios
54
55     // Controle o laço de repeticao
56     // Execute n repeticoes
57
58     // Demais instrucoes
59
60     return 0;
61 }

```

Entrada

Seu programa deve ler um inteiro correspondente à quantidade de polinômios a serem lidos. Em seguida, para cada polinômio da sequência, deverá ler a ordem seguido dos seis coeficientes.

Saída

O programa deve apresentar, para cada par de polinômios os três polinômios resultantes da soma, subtração e multiplicação. Os pares são formados sempre pelos polinômios de índice i e $i + 1$, ou seja, o primeiro forma par com o segundo, o segundo com o terceiro e assim por diante.

Exemplo

Entrada	Saída
3	soma: +3.0x ¹ +1.0x ⁰
1 1 -1	subtracao: -1.0x ¹ -3.0x ⁰
1 2 2	multiplicacao: +2.0x ² +0.0x ¹ -2.0x ⁰
1 1 2	soma: +3.0x ¹ +4.0x ⁰
	subtracao: +1.0x ¹ +0.0x ⁰
	multiplicacao: +2.0x ² +6.0x ¹ +4.0x ⁰