Prova-03

Prof. Dr. Gustavo Teodoro Laureano
Prof. Dr. Thierson Rosa Couto

Sumário

1 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 .

```
typedef struct {
   int numJogo;
   int numero[6];
} 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 palpite 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 o apostador acertar 4 números.
2. QUINA jogo: a b c d e: quando o apostador acertar 5 números.
3. SENA jogo: a b c d e f: quando o 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 n".

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

2 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 polinônio 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 polinônio $2x^3 - 1x^2 + 1$ é: `ordem=3` e `coef={1,0,-1,2}`.

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

Neste exercício, a impressão de um polinômio segue o seguinte padrão: $s_c c_p \hat{} p_c$, 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.

```c
typedef struct {
   int ordem;      // Ordem do polinomio
   double * coef; // Vetor de coeficientes
} Poly;

/**
* Funcao que cria um polinomio com alocacao dinamica
* @param Poly* Ponteiro para o novo polinomio
*/
Poly * poly_new(int ord);

/**
* Funcao que imprime um polinomio na tela
* @param p Ponteiro para o polinomio
*/
void poly_print(Poly * p);
/**
* Funcao que libera a memoria alocada a um polinomio
* @param p Ponteiro para o polinomio
*/
void poly_free(Poly * p);

/**
* Cria o polinomio resultante da soma
* @param A Ponteiro para o primeiro polinomio
* @param B Ponteiro para o segundo polinomio
* @return Poly* <- A + B
*/
Poly * poly_sum( Poly * A, Poly * B );

/**
* Cria o polinomio resultante da subtracao
* @param A Ponteiro para o primeiro polinomio
* @param B Ponteiro para o segundo polinomio
* @return Poly* <- A - B
*/
Poly * poly_sub( Poly * A, Poly * B );

/**
```

```
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 laco 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 ses coeficientes.

Saída

O programa deve apresantar, 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 1 1 -1 1 2 2 1 1 2	soma: +3.0x^1+1.0x^0 subtracao: -1.0x^1-3.0x^0 multiplicacao: +2.0x^2+0.0x^1-2.0x^0 soma: +3.0x^1+4.0x^0 subtracao: +1.0x^1+0.0x^0 multiplicacao: +2.0x^2+6.0x^1+4.0x^0