

Atividade: Prévia 2 da Prova 02

Professor: Hebert Coelho da Silva

Disciplina: Introdução a Programação

Leia atentamente as instruções

Observações:

- Todas as questões devem ser respondidas usando a Linguagem C.
- As questões devem ser enviadas pelo sistema sharif.inf.ufg.br/hebert.
- A prova deve ser feita individualmente.
- É proibido consultar a Internet, livros, anotações ou qualquer outro material durante a prova.
- A correção automatizada só funciona quando todas as exigências da tarefa quanto ao formato da entrada e saída de seu programa estiverem corretas.
- Não implemente nenhum recurso gráfico nas suas soluções (janelas, menus, etc.), nem utilize qualquer rotina para limpar a tela ou posicionar o cursor.
- As tarefas não estão ordenadas, neste caderno, por ordem de dificuldade; procure resolver primeiro as questões mais fáceis.
- Todos os dados devem ser lidos da entrada padrão (normalmente é o teclado) e escritos na saída padrão (normalmente é a tela). Utilize as funções padrão para entrada e saída de dados: **scanf**, **getchar**, **printf**, **putchar**, **puts**, **gets**; Para ler uma string preferencialmente use a função **scanf("%[^\n]", nome_da_variavel)**. Existe um espaço entre " %" no último comando.
- **Utilize funções para resolver todas as questões.**

Número Primo na Matriz

Nome do arquivo fonte: pMatriz.c

O aluno JOY juntamente com a aluna RIBECA perceberam que uma tarefa muito fácil poderia cair na prova de Álgebra Linear e resolveram tentar respondê-la. A tarefa é encontrar o primeiro número primo "P" diferente de 1 e 2 na diagonal principal de uma matriz "A" de dimensão NxN, em seguida multiplicar os elementos de cada coluna com paridade diferente da coluna onde P se encontra pelo valor de P. A ordem dos elementos na diagonal principal é definida pelos pares ordenados (1,1), (2,2), (3,3), e assim por diante.

Caso não exista elemento primo diferente de 1 e 2 na diagonal principal da matriz A, pegar o último elemento da diagonal principal e multiplicar por cada elemento das linhas ímpares da matriz A. Considerar uma linha par ou ímpar pela posição em linguagem C, ou seja, linha zero é par, linha 1 é ímpar e assim por diante.

Ao final mostrar o valor da matriz A após realizada as operações.

O seu programa não deverá usar variáveis globais. Fazer uso de 4 funções, todas usam como parâmetro de entrada a matriz de dimensão máxima 500x500, e a dimensão da matriz.

A primeira função além dos 2 parâmetros citados, recebe um ponteiro para inteiro que indica se achou primo 1 ou não achou primo 0, função retorna pelo seu nome a posição do número primo na diagonal principal ou a última posição caso não exista primo na diagonal principal.

Protótipo da função 1:

```
int PosPrimoDiagPMatriz(int M[500][500], int N, int *TemPrimo); //return posição
```

A segunda função altera a matriz recebida na entrada caso encontre um número primo.

A terceira função altera a matriz recebida na entrada caso não encontre um número primo.

A quarta função imprime a matriz recebida.

Entrada

A primeira linha contém um inteiro $N < 501$, indicando a dimensão da matriz A.

As N linhas seguintes contém os elementos da matriz A.

Saída

Seu programa deve produzir N linhas e N colunas com o valor da matriz A alterado.

Exemplos

| Entrada 1 | Saída 1 |
|---------------|----------------|
| 5 | |
| 1 2 3 4 5 | 3 2 9 4 15 |
| 2 3 4 5 6 | 6 3 12 5 18 |
| 7 8 9 10 11 | 21 8 27 10 33 |
| 8 9 10 11 12 | 24 9 30 11 36 |
| 9 10 11 12 13 | 27 10 33 12 39 |
| Entrada 2 | Saída 2 |
| 3 | 1 2 3 |
| 1 2 3 | 8 4 16 |
| 2 1 4 | 5 3 4 |
| 5 3 4 | |

Nomes

Nome do arquivo fonte: nomes.c

A garota Notielc trabalha com nomes todo tempo e deseja saber se algum nome ao contrário forma um nome conhecido por ela.

Você deverá fazer um programa que dado um conjunto de nomes escritos por Notielc, imprimir os nomes ao contrário e em ordem alfabética, a ordem alfabética deverá ser considerada com os nomes ao contrário. **Não usar função de ordenação que já esteja pronta na linguagem de programação, por exemplo qsort(). Considere que a ordem alfabética é a mesma dada pela tabela ASCII. Se possível, utilize 3 funções, uma que inverte a lista de nomes, outra que ordena os nomes e por último uma que imprime a lista de nomes.**

Entrada

A primeira linha possui dois inteiros $N < 101$ e $M < 501$, N é a quantidade de nomes que Notielc irá digitar e M é a quantidade máxima de letras que cada nome deverá ter.

Saída

Seu programa deve produzir N linhas, contendo os nomes ao contrário em ordem alfabética. As letras nos nomes da saída tem letras minúsculas e maiúsculas conforme aparecem na entrada.

Exemplos

| Entrada | Saída |
|---------|---------|
| 4 10 | alisoN |
| Notielc | anA |
| Nosila | artuR |
| Rutra | cleitoN |
| Ana | |

Estudar não dói

Nome do arquivo fonte: doi.c

Isla vive escutando da sua mãe que estudar não dói. Atualmente, Isla entrou na Universidade para cursar Ciência da Computação e esta tendo uma disciplina que é muito difícil, assim ele não esta concordando muito com sua mãe. Nesta disciplina Isla tem que fazer muitas manipulações com conjuntos. Devido ao tamanho dos conjuntos Isla passa horas para testar propriedades simples. Neste momento Isla esta verificando se elementos de dois conjuntos A e B são pares ou ímpares. Sua tarefa é ajudar Isla a encontrar os elementos pares do conjunto A e encontrar os elementos ímpares do conjunto B e formar um novo conjunto C com tais elementos. O conjunto C deverá conter a concatenação entre os pares de A e ímpares de B, ou seja, um elemento par do conjunto A, depois um elemento ímpar do conjunto B, depois o próximo elemento par do conjunto A e assim por diante. Se os elementos do conjunto A terminarem primeiro que os elementos do conjunto B, então as últimas posições de C só terão elementos ímpares de B. Caso os elementos de B terminem antes dos elementos de A, então as últimas posições de C só terão elementos pares de A. **Escreva no mínimo uma função que recebe os dois vetores com uso de ponteiros e concatene os seus valores no vetor C.**

Protótipo da função:

```
void Concatena(int *A, int *B, int *C, int *TamC); //A, B, C vetores e TamC inteiro.
```

Outras funções podem ser para imprimir a matriz C, para gerar um vetor com os elementos pares de A, para gerar um vetor com os elementos ímpares de B.

Entrada

A primeira linha possui dois inteiros $N < 5001$ e $M < 5001$, N é a quantidade de elementos do conjunto A e M é a quantidade de elementos do conjunto B. A segunda linha contém os elementos do conjunto A. A terceira linha contém os elementos do conjunto B.

Saída

Seu programa deve produzir o conjunto C, com os elementos pares do conjunto A e os elementos ímpares do conjunto B. Após o último valor deverá haver um espaço.

Exemplos

| | |
|--|---------------------------------------|
| Entrada 1 4 10 1 2 3 4 5 7 8 9 0 6 11 14 13 10 | Saída 1 2 5 4 7 9 11 13 |
| Entrada 2 5 3 6 1 3 11 13 5 7 9 | Saída 2 6 5 7 9 |