



Fundação CECIERJ - Vice-Presidência de Educação Superior a Distância

Curso de Tecnologia em Sistemas de Computação

Disciplina Fundamentos de Programação

AP3 1º semestre de 2017

IMPORTANTE

- Prova sem consulta e sem uso de qualquer aparato eletrônico.
- Use caneta para preencher o seu nome e assinar nas folhas de questões e nas folhas de respostas.
- Você pode usar lápis para responder as questões.
- Ao final da prova, devolva as folhas de questões e as de respostas.
- Todas as respostas devem ser transcritas nas folhas de respostas. As respostas nas folhas de questões não serão corrigidas.

1ª Questão (3,5 pontos)

Faça um programa que receba do usuário, via entrada padrão, um par de números inteiros, denominados *menor* e *maior*, respectivamente, e escreva na saída padrão todos os números naturais primos contidos no intervalo definido por esse par. O intervalo é fechado. Logo, *menor* e *maior* fazem parte do intervalo.

Um número inteiro P é dito *natural primo* se e somente se P é inteiro, maior que 1 e é apenas divisível por 1 e por ele mesmo, P , sem deixar resto. Por exemplo, 5 é um número primo porque dentre os números 1, 2, 3, 4 e 5 apenas o 1 e 5 resultam em divisão sem resto. Qualquer número maior que 5 claramente deixará resto. Já o número 4 não é primo, pois dentro os números 1, 2, 3 e 4 os números 1, 2 e 4 dividem 4 sem resto. Qualquer número maior que 4 claramente deixará resto.

Entrada

A entrada é composta por uma linha contendo um par de números inteiros positivos. É garantido que o primeiro número do par será menor ou igual ao segundo número.

Saída

A saída deve apresentar todos os números naturais primos contidos no intervalo informado, apresentado em cada linha um número natural primo identificado.

Exemplos

Entrada	Saída
1 10	2 3 5 7

Entrada	Saída
9 14	11 13

Entrada	Saída
8 10	

Obs: Nada foi impresso na saída, pois nenhum dos valores no intervalo (ou seja: 8, 9 e 10) é um número natural primos.

2ª Questão (3,0 pontos)

Escreva um programa que contenha uma função recursiva que calcule o número de grupos distintos com K pessoas que podem ser formados a partir de um conjunto com N pessoas. Ou seja, a função deverá calcular a quantidade possível de combinações sem repetição.

A definição recursiva para cálculo de combinações é apresentada abaixo:

$$Comb(n, k) = \begin{cases} n, & \text{se } k = 1 \\ 1, & \text{se } k = n \\ Comb(n-1, k-1) + Comb(n-1, k), & \text{se } 1 < k < n \end{cases}$$

Observação: A implementação apresentada deve ser recursiva e a recursão deve estar de acordo com a definição acima. Implementações não recursivas da função solicitada não serão consideradas na correção.

Entrada

A entrada é composta por uma linha contendo um par de números inteiros positivos, respectivamente, N e K , onde $1 \leq K \leq N$.

Saída

A saída deve apresentar o número de combinações possíveis.

Exemplos

Entrada	Saída
5 2	10

Entrada	Saída
10 3	120

3ª Questão (3,5 pontos)

Considere a existência de dois arquivos binários chamados “entrada1.bin” e “entrada2.bin”. Cada um desses arquivos armazena uma coleção já ordenada de valores inteiros.

Escreva um programa que leia o conteúdo desses dois arquivos e crie um novo arquivo binário, chamado “saida.bin”, que armazene a coleção que mescla o conteúdo de “entrada1.bin” e “entrada2.bin” mantendo a ordenação.

Entrada

A entrada consiste em um par de arquivos binários chamados “entrada1.bin” e “entrada2.bin”. Os primeiros 4 bytes de cada arquivo armazenam um valor inteiro N que indica quantos valores inteiros (cada valor composto por 4 bytes) existem no restante do arquivo. Os N valores já se encontram ordenados de forma crescente.

Saída

A saída consiste na criação de um arquivo binário chamado “saida.bin”. Esse arquivo segue o mesmo formato interno que os arquivos de entrada. Ou seja, os primeiros 4 bytes armazenam um valor inteiro M que indica quantos valores inteiros (cada valor composto por 4 bytes) existem no restante do arquivo. Os M valores deverão estar ordenados de forma crescente.

Exemplo

Se os arquivos de entrada fossem compostos pelos seguintes dados

entrada1.bin

6	10	15	20	22	23	40
---	----	----	----	----	----	----

entrada2.bin

3	8	16	60
---	---	----	----

então o arquivo de saída deveria conter a seguinte composição

saida.bin

9	8	10	15	16	20	22	23	40	60
---	---	----	----	----	----	----	----	----	----

Observação:

Os arquivos são binários. A entrada e saída de programas que consideram arquivos texto não serão considerados na correção.

Restrição:

Não é permitido manter todo o conteúdo do arquivo na memória principal, pois os arquivos são tão grandes que certamente levará à falta de memória e término prematuro do programa. Portanto, não utilize lista ou estruturas similares para manter todo o conteúdo do arquivo na memória principal. Códigos que não cumprem essa restrição serão considerados incorretos.

Boa Avaliação!