

Prova 1

Algoritmos e Estruturas de Dados I

Professor: Pedro O.S. Vaz de Melo

23 de setembro de 2014

Nome:

escrevendo o meu nome eu juro que seguirei o código de honra

Código de Honra para este exame:

- Não darei ajuda a outros colegas durante os exames, nem lhes pedirei ajuda;
- não copiarei nem deixarei que um colega copie de mim;
- não usarei no exame elementos de consulta não autorizados.

Informações importantes:

- Em questões que pede um **programa**, este deve ser completo, com bibliotecas (incluindo, quando necessário, a biblioteca `prova1.h`), função `main`, etc. Se deve ser feita uma **função**, somente a função é suficiente. Se deve ser feito um **procedimento**, somente o procedimento é suficiente.
- A interpretação das questões da prova faz parte do critério de avaliação. Caso tenha dúvida sobre a sua interpretação de uma determinada questão, escreva as suas suposições na resolução da mesma.
- As funções implementadas no módulo `prova1.h` podem ser usadas em **qualquer** exercício da prova. Além disso, se você usar uma função do módulo `prova1.h`, considere que ela está implementada de forma correta.

Referências:

Função/Operador	Descrição	Biblioteca	Exemplo
<code>%</code>	retorna o resto da divisão	-	<code>20 % 3</code> retorna 2

1. (8 points) Para as questões a seguir, considere que as implementações serão feitas no módulo "prova1.h".

a. (4 pts) Escreva uma **função** de nome `primosEntreSi` que verifica se dois números inteiros x e y são primos entre si. Se os números forem primos entre si a função deve retornar 1 e, caso negativo, deve retornar 0. Dois números são primos entre si se o máximo divisor comum entre eles for 1. Assim, a função `primosEntreSi` deve retornar 1 se os parâmetros de entrada forem 27 e 32 e 0 se os parâmetros de entrada forem 26 e 32. Essa função deve ter o seguinte protótipo:

```
int primosEntreSi(unsigned int x, unsigned int y);
```

b. (4 pts) Escreva um **procedimento** de nome `multiplicaSePrimosEntreSi` que recebe como parâmetro dois endereços de memória de variáveis inteiras `end_var1` e `end_var2`. A função deve verificar se o conteúdo desses endereços de memória são primos entre si. Caso positivo, a função deve multiplicar os dois valores e armazenar o resultado em ambos endereços de memória. Caso negativo, a função deve armazenar o valor 0 nos dois endereços de memória.

2. (5 points) Escreva um **programa** que lê dois números inteiros do teclado: x_{min} e x_{max} e imprime na tela todos os pares **distintos** de números naturais que são primos entre si e que estão entre x_{min} e x_{max} (inclusive para ambos). Enquanto o usuário inserir um valor negativo para x_{min} ou x_{max} ou enquanto x_{min} for maior ou igual a x_{max} , o programa deve pedir a ele novos valores de x_{min} e x_{max} . Exemplo: se $x_{min} = 3$ e $x_{max} = 7$, o programa deve imprimir (3, 4), (3, 5), (3, 7), (4, 5), (4, 7), (5, 6), (5, 7), (6, 7). Note que o par (4, 3) não deve ser impresso, pois o par (3, 4) já foi impresso. Obs: neste programa você não precisa imprimir qualquer mensagem para o usuário, apenas a resposta final.

3. (3 points)

a. (2 pts) Complete o código abaixo, considerando que as variáveis x e y vão ser usadas como parâmetros nas linhas 7, 8, 9 e 11:

```
1: #include <stdio.h>
2: #include _____
3:
4: void main(void) {
5:     int x,y;
6:     printf("Digite os valores de x e y\n");
7:     scanf(_____);
8:     multiplicaSePrimosEntreSi(_____);
9:     printf("Os novos valores de x e y sao: x=_____ e y=_____\n", _____, _____);
10:    multiplicaSePrimosEntreSi(_____);
11:    printf("Os novos valores de x e y sao: x=_____ e y=_____\n", _____, _____);
12: }
```

b. (1 pt) O que foi impresso nas linhas 9 e 11 caso o usuário tenha entrado com os valores $x = 4$ e $y = 7$ na linha 7?