1ª Avaliação Individual
Estruturas de Dados I
Prof. Leandro M. Zatesko
19 de março de 2015

- Este instrumento avaliativo tem início às 19:20 e fim às 20:50. Durante e apenas durante este tempo, o estudante deve resolver as questões, colocando suas soluções num único arquivo de nome `NomeCompletoDoEstudante.zip` na Área de Trabalho, sendo `NomeCompletoDoEstudante` o nome completo do estudante sem diacríticos nem espaços (e.g. o arquivo de Ciça Guimarães deve se chamar `CicaGuimaraes.zip`).
- Esta folha de questões possui 2 páginas (frente e verso).
- Nenhum material pode ser consultado, embora todos os programas e aplicativos instalados no ambiente que não façam uso da Internet possam ser usados. Apenas as máquinas do Laboratório podem ser usadas.
- Caso o estudante deseje fazer uso das folhas de rascunho, deve solicitar ao professor.
- Sobre a mesa são permitidos apenas lápis, canetas e borrachas. Quaisquer outros objetos, como estojos, capacetes, bolsas e mochilas, devem ser acomodados no chão ou na prateleira sobre a mesa.
- O estudante que precisar ir ao banheiro poderá fazê-lo apenas solicitando ao professor. Contudo, não será permitido que mais de um estudante esteja ausente do Laboratório ao mesmo tempo.
- Após terminar sua avaliação, o estudante deverá, permanecendo sentado, solicitar ao professor a submissão de seu arquivo `.zip`.
- Para operações de leitura e impressão, seus códigos devem sempre considerar os dispositivos padrões de entrada e saída (`stdin` e `stdout`, respectivamente).
- O estudante que tentar de algum modo *hackear* o bloqueio da Internet, reiniciar sua máquina ou sua sessão ou fraudar o instrumento avaliativo de algum modo, ou que descumprir alguma das regras estabelecidas neste cabeçalho, terá sua nota imediatamente anulada.

Questão 1 (2 pontos)**.** Todo número racional positivo pode ser expresso de forma única como uma fração irredutível. Escreva um programa em C que leia um número racional q na forma decimal com exatas 3 casas decimais e imprima o mesmo número na forma de uma fração irredutível. Considere que $0 < q \leqslant 10^6$ e que o separador das casas decimais é o ponto (`.`), não a vírgula (`,`). Imprima a fração numa única linha, separando o numerador do denominador apenas por uma barra (`/`). Não se esqueça de finalizar a linha da saída com uma quebra de linha (`'\n'`). Por exemplo, se seu programa ler:

```
14.750
```

deverá imprimir:

```
59/4
```

Nomeie seu código como `questao1.c` (não como `questao1.cpp`) no seu `.zip`.

Questão 2 (2 pontos)**.** Ordene as funções abaixo de modo que, se uma função f aparecer antes de uma função g na ordem que você estabelecer, é porque $f \ll g$.

$$n \quad 2^n \quad 2^{2n} \quad n! \quad n\log n \quad n^2 \quad 2^{n^2} \quad \log n \quad 3^n \quad \sqrt{n}$$

Você deve apenas apresentar a ordenação, não sendo necessário justificá-la. Apresente sua resposta num arquivo nomeado `questao2.pdf` no seu `.zip`.

Questão 3 (2 pontos)**.** Se a Conjectura de Goldbach for verdadeira, todo número par maior que 2 pode ser expresso como a soma de dois números primos (não necessariamente distintos). Se a conjectura não for verdadeira, não o será para algum número muitíssimo grande. Para números pequenos, como números que cabem em 32 ou 64 *bits*, já sabemos que a conjectura vale. Porém, nem sempre a soma da qual a conjectura trata é única. Para os números 4 ou 8, a soma é única, pois $4 = 2 + 2$ e $8 = 3 + 5$, mas para 42, por exemplo, há quatro possibilidades para a soma: $42 = 5 + 37 = 11 + 31 = 13 + 29 = 19 + 23$.

Podemos dizer que $5+37$ é a *primeira* soma de dois primos que expressa 42 porque a diferença entre os primos que constituem a soma é a maior possível (veja que essa diferença é 32, enquanto que nas outras somas é 20, 16 ou 4).

Escreva um programa em C que leia um inteiro positivo par n ($2 < n \leqslant 10^7$) e imprima os primos p_1 e p_2 tais que $p_1 + p_2 = n$ e tais que a diferença entre p_1 e p_2 é máxima, ou seja, tais que $p_1 + p_2$ é a *primeira* soma de primos que expressa n. Seu programa deverá, ao receber n, usar o Crivo de Eratóstenes para gerar todos os primos menores que n e então procurar pelo primeiro primo p_1 tal que $n - p_1$ seja também primo. Na saída, separe os primos por um espaço em branco apenas, não se esquecendo de finalizar a linha com uma quebra de linha (`'\n'`). Por exemplo, se seu programa ler:

```
42
```

deverá imprimir:

```
5 37
```

Nomeie seu código como `questao3.c` (não como `questao3.cpp`) no seu `.zip`.

Questão 4 (2 pontos)**.** Escreva uma função *recursiva* em C de protótipo

```
int udnndf(int n);
```

que, ao receber um inteiro n ($0 \leqslant n < 2^{31}$), devolve o último dígito não-nulo do fatorial de n. Por exemplo, ao receber 10, sua função deverá devolver 8, pois $10! = 3628800$. Já, ao receber 20, sua função deverá devolver 4, pois $20! = 2432902008176640000$. Seu código, nomeado como `questao4.c` (não como `questao4.cpp`), deverá conter apenas a função `udnndf()` e estar presente no seu `.zip`.

Questão 5 (2 pontos)**.** A seguinte função recebe duas matrizes quadradas de inteiros A e B de dimensões $n \times n$ ($1 \leqslant n \leqslant$ MAX) e escreve na matriz C o resultado da multiplicação de A por B.

```
void multmat(int n, int A[MAX][MAX], int B[MAX][MAX], int C[MAX][MAX]) {
  int i, j, k;
  for (i = 0; i < n; i++)
    for (j = 0; j < n; j++) {
      C[i][j] = 0;
      for (k = 0; k < n; k++)
        C[i][j] += A[i][k] * B[k][j];
    }
}
```

Escreva uma função em C de protótipo

```
void expmat(int n, int A[MAX][MAX], int k, int B[MAX][MAX]);
```

que recebe uma matriz quadrada de inteiros A de dimensões $n \times n$ ($1 \leqslant n \leqslant$ MAX) e escreve na matriz B o resultado da elevação de A à k-ésima potência. Sua função deverá ser recursiva, usar Exponenciação Binária e, para o cálculo da multiplicação entre matrizes, chamar a função `multmat`. Você não deve se preocupar com *overflow* ou com a definição da constante MAX. Você não deve se preocupar nem mesmo com copiar a função `multmat`. Tudo o que seu arquivo `questao5.c` (não `questao5.cpp`) deve conter é a função `expmat()`, precedida por um comentário contendo a complexidade de tempo em função de n e de k, usando a notação O. Não é necessário justificar a complexidade, basta somente apresentá-la.