Departamento de Ciência da Computação – IME – USP

MAC2166 – Introdução à Ciência da Computação

Escola Politécnica – Computação / Elétrica – Primeiro Semestre de 2023

Exercício-Programa 1-b (EP1-b)
Data de Entrega: **30 de abril**

Para se preparar bem para o desenvolvimento de seu EP1-b, cuja descrição se inicia na próxima página, leia com atenção as instruções abaixo.

- Utilize **somente** os recursos da linguagem que aprendeu nas aulas.
- Veja em `https://www.ime.usp.br/~mac2166/infoepsC/` as instruções de entrega dos exercícios-programa e atente para as instruções de preenchimento do cabeçalho do seu programa.
- Caso você tenha dúvidas sobre eventuais erros e *warnings* que o compilador produza ao processar o seu programa, consulte o FAQ sobre compilação em `https://www.ime.usp.br/~mac2166/compilacao/`.
- Sempre compile seus programas com as opções **-Wall -ansi -pedantic -O2**. Seu programa deve:
  - funcionar para qualquer entrada que está de acordo com o enunciado (não é necessário verificar se a entrada está "bem formada");
  - estar em conformidade com o enunciado;
  - estar bem estruturado;
  - ser de fácil compreensão, com o uso padronizado da linguagem C.

Lembre-se de que o EP1-b valerá 9.0 (nove) pontos. Sua nota no EP1 será a soma de suas notas no EP1-a e EP1-b.

EP1-b: Uma base diferente

Em um livro publicado em 1202, o matemático Leonardo Fibonacci considerou uma sequência de números inteiros que passou a ser conhecida como *sequência de Fibonacci*. Para os nossos propósitos, a sequência começa pelos números 1 e 2, e cada termo subsequente é a soma dos dois anteriores:

1 2 3 5 8 13 21 34 55 89 144 233 377 610 987 1597 2584 $\cdots$

Formalmente, definimos $F_1 = 1$, $F_2 = 2$, e $F_k = F_{k-1} + F_{k-2}$ para todo $k \geq 3$.

A sequência de Fibonacci possui uma propriedade interessante que é expressa pelo seguinte teorema:

> **Teorema:** Todo inteiro positivo pode ser representado de uma maneira única como uma soma de números de Fibonacci distintos e sem dois números de Fibonacci consecutivos. Precisamente, dado um inteiro positivo N, existe uma única sequência binária $b_t b_{t-1} \ldots b_1$ com $b_t \neq 0$ e sem dois 1s consecutivos tal que $N = \sum_{i=1}^{t} b_i F_i$.

Dizemos que a sequência binária $b_t b_{t-1} \ldots b_1$ é a *representação Fibonacciana* de N. Por exemplo, $62 = 55 + 5 + 2$, logo a representação Fibonacciana de 62 é 100001010.

Seu programa

Neste exercício-programa você deve escrever um programa em C que lê um número `opcao`, que vale 0, 1, ou 2, seguido de uma sequência de inteiros positivos cujo término é indicado por um 0, e que executa uma das seguintes tarefas dependendo do valor de `opcao`:

0: converte os números da sequência para a representação Fibonacciana;

1: converte os números da sequência em somas de números de Fibonacci distintos, sem dois números de Fibonacci consecutivos;

2: converte os números da sequência, que estão em representação Fibonacciana, para o inteiro correspondente.

Os exemplos a seguir devem deixar claro o que se pede.

Considere os arquivos de entrada `entrada0.txt`, `entrada1.txt` e `entrada2.txt`, disponíveis no e-Disciplinas. Seu programa poderia ser executado com os dados nesses arquivos da seguinte forma, em que `ep1b` é o arquivo executável gerado de seu código em C:

```
$ ./ep1b < entrada0.txt
100001010
1010000010
1010101010
1001010010100001
10010100100101001000010101010001010
```

```
100010001001000001010000101010000000100001
$ ./ep1b < entrada1.txt
F_9 (55) + F_4 (5) + F_2 (2)
F_10 (89) + F_8 (34) + F_2 (2)
F_10 (89) + F_8 (34) + F_6 (13) + F_4 (5) + F_2 (2)
F_16 (1597) + F_13 (377) + F_11 (144) + F_8 (34) + F_6 (13) + F_1 (1)
F_35 (14930352) + F_32 (3524578) + F_30 (1346269) + F_27 (317811) + F_24 (75025) + F_22 (28657)
F_41 (267914296) + F_37 (39088169) + F_33 (5702887) + F_30 (1346269) + F_24 (75025) + F_22 (2865
$ ./ep1b < entrada2.txt
7
20
62
125
143
```

Note que no caso da execução com entrada `entrada1.txt`, a saída acima está "cortada". As saídas acima estão nos arquivos `saida0.txt`, `saida1.txt` e `saida2.txt`, disponíveis no e-Disciplinas. Esses arquivos foram gerados da seguinte forma:

```
$ ./ep1b < entrada0.txt > saida0.txt
$ ./ep1b < entrada1.txt > saida1.txt
$ ./ep1b < entrada2.txt > saida2.txt
```

Vamos agora discutir os exemplos acima em mais detalhes.

Conversão para a base Fibonacciana

O conteúdo de `entrada0.txt` é como segue:

```
0
62
125
143
2166
20230430
314159265
0
```

Note que essa entrada especifica `opcao` 0, isto é, seu programa deve converter cada um dos números 62, 125, 143, ... para a base Fibonacciana. Note que a saída `saida0.txt` começa com 100001010, que é 62 na base Fibonacciana, e termina com

$$100010001001000001010000101010000000100001,$$

que é 314159265 na base Fibonacciana.

A saída de seu programa executado com entrada `entrada0.txt` deve ser **exatamente** como em `saida0.txt`.

Conversão para soma de números de Fibonacci

O conteúdo de `entrada1.txt` é como segue:

```
1
62
125
143
2166
20230430
314159265
0
```

Note que essa entrada especifica `opcao` 1, isto é, seu programa deve converter cada um dos números 62, 125, 143, ... para uma soma de números de Fibonacci sem repetição e sem dois números de Fibonacci consecutivos. Note que a saída `saida1.txt` começa com

```
F_9 (55) + F_4 (5) + F_2 (2)
```

A saída de seu programa executado com entrada `entrada1.txt` deve ser **exatamente** como em `saida1.txt`.

Conversão da base Fibonacciana

O conteúdo de `entrada2.txt` é como segue:

```
2
1010
101010
100001010
1010000010
1010101010
0
```

Note que essa entrada especifica `opcao` 2, isto é, seu programa deve converter 1010, 101010, 100001010, ... para a base decimal. Note que a saída `saida2.txt` começa com 7, 20, ..., dado que $7 = 5 + 2 = F_4 + F_2$, $20 = 13 + 5 + 2 = F_6 + F_4 + F_2$, etc.

A saída de seu programa executado com entrada `entrada2.txt` deve ser **exatamente** como em `saida2.txt`.

Observação. Para tornar esta parte do exercício compatível com o que sabemos da linguagem C até o momento, vamos supor que, com `opcao` 2, o maior número que será dado como entrada é, na base Fibonacciana, 1010101010.

Testes para seu programa

No e-Disciplinas disponibilizamos alguns arquivos adicionais de entrada e saída. Experimente executar seu programa com esses arquivos de entrada, e verifique se a saída de seu programa é como nos arquivos de saída correspondentes.

Para testar seu programa com o arquivo de entrada `entradaN.txt`, compile seu programa e produza um arquivo de saída como abaixo:

```
$ gcc ep1b.c -Wall -ansi -pedantic -O2 -o ep1b
$ ./ep1b < entradaN.txt > saidaN_minha.txt
```

Compare a saída produzida pelo seu programa (`saidaN_minha.txt`) com o arquivo de saída `saidaN.txt` disponibilizado (por exemplo, carregando esses arquivos no editor do VS Code). Certifique-se de que o conteúdo de `saidaN_minha.txt` coincide com o conteúdo de `saidaN.txt`. Existem jeitos simples e convenientes de se comparar o conteúdo de dois arquivos (utilitários `diff`, `md5sum` ou `md5`); pergunte sobre eles em sala ou nas monitorias.