MAC 115 – Introdução à Ciência da Computação

Primeiro Exercício–Programa

Nossos Caminhos se Cruzarão!

1 Introdução

Começamos descrevendo um jogo de baralho. Os nossos personagens são o *Professor Teoria* e o *Aluno Brilhante*. O prof. Teoria mostra o seguinte truque de baralho para o Aluno Brilhante. Inicialmente, ele pede ao aluno que embaralhe bastante um jogo de baralho. Então, o prof. Teoria pede ao aluno que siga as seguintes instruções:

(1) Escolha um número entre um e meia dúzia, digamos n_0.

(2) Começando com todo o jogo de baralho fechado, abra as cartas uma a uma e veja qual é o valor numérico da n_0-ésima carta (isto é, você ignora o naipe, olha apenas para o número).[1] Por exemplo, se $n_0 = 4$, o Aluno Brilhante observará a quarta carta do baralho aberta. Supondo que o valor desta carta seja n_1, temos, claramente, $1 \leq n_1 \leq 13$.

(3) Continue abrindo as cartas, uma a uma, até chegar na n_1-ésima carta, *contando a partir da carta anterior*. Por exemplo, se $n_0 = 4$ e $n_1 = 3$, o Aluno Brilhante deverá chegar na $(n_0 + n_1)$-ésima carta neste instante, isto é, na sétima carta do baralho, contando a partir da primeira.

(4) A partir daqui, o Aluno Brilhante deve repetir o processo do item (3) acima, isto é, ele deve observar o valor numérico da carta na qual ele chegou, deve então contar este número de cartas a partir daquela carta, observar a carta que aparece, contar a partir dela, e assim por diante.

(5) Naturalmente, uma hora o baralho se esgota. Neste ponto, o Aluno Brilhante deve lembrar a última carta que ele observou no processo.

O prof. Teoria então pensa um pouco, usa toda a teoria que ele sabe, e indica ao Aluno Brilhante qual carta ele acha que foi a última carta que o Aluno Brilhante observou neste processo. Naturalmente, o prof. Teoria acerta! (Na maioria das vezes.)

Vamos considerar um exemplo. Suponha que as cartas que aparecem ao abrirmos as cartas uma a uma são

```
5 6 10 3 12 7 7 4 8 9 10 3 10 4 ... 12 1 4 2 2 11 6 10 4 8 1 13 7 11
```

(Os naipes não estão indicados, já que nós os ignoramos neste truque. Também omitimos algumas cartas no meio.) Suponha, agora, que o Aluno Brilhante escolheu $n_0 = 4$. Então, no processo acima, ele vai observar as cartas marcadas com `'+'` como segue:

```
5 6 10 3+ 12 7 7+ 4 8 9 10 3 10 4+ ... 12 1+ 4+ 2 2 11 6+ 10 4 8 1 13 7+ 11
```

Nesta situação, o prof. Teoria deve acertar que a última carta observada pelo Aluno Brilhante foi a penúltima carta, de número `7`.

Note que se o prof. Teoria consegue adivinhar o valor de n_0, é claro que ele consegue acertar a carta final do Aluno Brilhante (basta ele seguir o procedimento (1)–(5), exatamente como o aluno faz). Entretanto, lembrando que o aluno tinha 6 escolhas para n_0, é improvável que o

[1] Como é usual, adotamos a convenção em que um Ás vale 1, um Valete vale 11, uma Rainha vale 12 e um Rei vale 13.

prof. Teoria consiga adivinhar este número com freqüência (no exemplo acima, o prof. Teoria teria de descobrir que o Aluno Brilhante começou a contar a partir da 4a. carta, isto é, $n_0 = 4$). A pergunta, então, é: como o prof. Teoria consegue adivinhar a *carta final* do Aluno Brilhante, mesmo que ele tenha pequenas chances de adivinhar a *carta inicial*?

(Pausa para pensar. Você deve tentar pensar qual é o truque do prof. Teoria antes de prosseguir esta intrigante leitura.)

Muito bem. Vamos abrir o jogo. O prof. Teoria faz o seguinte: ele faz a mesma coisa que o aluno, começando a contar a partir de qualquer carta! Na verdade, ele até faz mais simples. Ele sempre escolhe contar a partir da primeira carta (é como se tivéssemos $n_0 = 1$). Naturalmente, se o aluno *não* escolheu $n_0 = 1$, não há porque as cartas do aluno e as cartas do professor terem qualquer relação. O interessante é que, mesmo assim, no fim do processo, o aluno e o professor estarão observando as mesmas cartas!

Vamos ilustrar isto no exemplo acima. Marquemos com `'*'` as cartas do prof. Teoria.

```
5* 6 10 3+ 12 7* 7+ 4 8 9 10 3 10* 4+ ... 12 1+ 4+ 2 2 11 6+ 10 4* 8 1 13 7+* 11
```

Neste exemplo, tanto o professor quanto o aluno têm 7 (sete) como carta final.

Experimente aplicar este truque! Algumas simulações indicam que a probabilidade de o prof. Teoria acertar é próximo de 75%. Nada mal, já que ele tem apenas uma chance em seis de acertar n_0!

2 O exercício-programa

A idéia deste exercício-programa é simular uma *variante* do jogo descrito acima. O motivo pelo qual simulamos uma variante e não este jogo em si é que, no momento, ainda não aprendemos todas as ferramentas necessárias para simular o processo de gerar uma 'ordenação aleatória das cartas de um baralho' (mais para o fim do semestre, você será capaz de fazer isto[2]).

A variante que simulamos neste exercício-programa é como segue. Ao invés de gerarmos uma seqüência de números entre 1 e 13, simulando uma ordenação aleatória das cartas de um baralho, vamos simplesmente gerar números inteiros entre 1 e 13 ao acaso. Na verdade, para termos maior generalidade, vamos gerar números entre `1` e `n_max` ao acaso, onde `n_max` é um inteiro positivo fornecido pelo usuário do programa (no caso de um baralho `n_max = 13`). Neste caso, podemos, por exemplo, gerar uma seqüência que começa com `1, 1, 1, 1, 1, 2, ...`, a qual, no caso de baralhos, seria *impossível.*

Para gerarmos números aleatórios, usaremos uma 'função da biblioteca padrão do `C`'. No momento, não se preocupe com o que isto significa exatamente. Você deve experimentar o seguinte programa:

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
  int n_max, N, semente, k, x;

  <leia n_max, N e semente>  /* Você irá completar esta parte em (A) abaixo */
```

[2] Se você já programou antes, pense como você faria isto!

```
  srand(semente);
  for (k = 0; k < N; k++) {
    x = 1 + (rand()/((double)RAND_MAX+1))*n_max;
    printf("%d ", x);
  }
  printf("\n");
  return 0;
}
```

O programa acima gera e imprime `N` inteiros aleatórios, todos entre `1` e `n_max`. Para gerar tais números, o programa acima usa uma certa 'função de biblioteca' chamada `rand()`. A *semente* é um número que esta função usa para gerar tais números. Discutiremos funções mais tarde no semestre.

Como exemplo, vamos considerar o caso em que `n_max = 7` e `N = 15`. Uma possível saída para o programa acima, dependendo da semente que você escolha, é

```
7 5 2 2 1 4 2 1 2 4 1 2 6 4 5
```

Considerando esta seqüência de números aleatórios gerada pelo nosso programa, podemos executar os passos do jogo de baralho discutido acima, com o prof. Teoria observando a primeira carta e o Aluno Brilhante observando a segunda carta (o número escolhido pelo prof. Teoria é 1, e o número escolhido pelo Aluno Brilhante é 2). Seguindo a mesma notação que antes (as cartas observadas pelo prof. Teoria estão marcadas com `'*'`, e as cartas observadas pelo Aluno Brilhante estão marcadas com `'+'`), temos

```
7* 5+ 2 2 1 4 2+ 1* 2*+ 4 1*+ 2*+ 6 4*+ 5
```

Note que, após gerarmos 9 números, já temos a coincidência entre os números do aluno e do professor! Eis um outro exemplo, com `n_max = 7` e `N = 15`:

```
4* 6+ 3 5 1* 3* 1 2+ 5* 4+ 1 3 7 2*+ 1
```

Ocorreu coincidência no 14o. número!

O objetivo principal deste exercício é escrever um programa para gerar saídas como as duas seqüências acima. Vamos agora descrever suas tarefas em detalhe.

3 O que se pede

Você deve escrever os seguintes programas para este exercício.

(A) *Programa preliminar.* Complete o programa dado acima, inserindo os comandos de leitura de `n_max`, `N`, e `semente`. Isto é apenas um aquecimento.

(B) *Programa principal.* Agora você deve "incrementar" o programa que você escreveu na etapa (A) para que ele também imprima os caracteres `'*'` e `'+'`, de acordo com o que foi discutido na Seção 2. *Assuma que o prof. Teoria sempre começa a contar da primeira carta e o Aluno Brilhante começa a contar da segunda carta.* Ademais, já que não há muito interesse em ver as seqüências após a coincidência entre os números do Aluno Brilhante e do prof. Teoria, *você deve fazer com que seu programa imprima os números exatamente até ocorrer tal coincidência.* Note que, em particular, neste programa você não deve ler `N`. Finalmente, o seu programa deve imprimir *o número de números que foram gerados até ocorrer a coincidência.* Portanto, a saída de seu programa deve ser algo como o seguinte:

```
5* 10+ 2 8 12 7* 5 3 3 8 5 1+ 13*+
Numero de numeros gerados = 13
```

ou o seguinte:

```
13* 6+ 9 5 7 8 2 12+ 4 10 4 4 3 10* 8 3 11 5 9 2+ 9 6+ 9 4* 10 13 13 11*+
Numero de numeros gerados = 28
```

4 Programa bônus

Talvez você fique curioso em saber, em média, quantos números você precisa gerar até ocorrer a coincidência entre os números do Aluno Brilhante e do prof. Teoria. Para tanto, você poderia executar o seu programa (B) acima várias vezes e tirar a média do número de números gerados em cada execução de seu programa. Uma forma melhor de fazer isto é, naturalmente, *incrementar o seu programa para que ele mesmo faça isto.* Neste exercício, você pode entregar ainda um terceiro programa que faz o seguinte.

(C) O seu terceiro programa deve receber como entrada mais um inteiro que indica quantas vezes simulamos o jogo acima. Note que, para tanto, o seu programa pode ter a seguinte estrutura:

```
[...]
int num_vezes, kk;
[...]
<leia n_max, semente e num_vezes>
[...]
for (kk = 0; kk < num_vezes; kk++) {
  <basicamente o seu programa de (B)>
}
[...]
printf("Numero medio de numeros gerados = %d/%d = %d\n",
        NN, num_vezes, NN/num_vezes);
[...]
```

onde `NN` é uma variável cujo valor é o número total de números gerados. Note também que tanto `n_max` como `semente` são lidos uma única vez. A saída de seu programa, neste caso, pode ter a seguinte forma:

```
Numero de numeros gerados = 28
Numero de numeros gerados = 103
Numero de numeros gerados = 24
Numero de numeros gerados = 7
Numero de numeros gerados = 36
Numero de numeros gerados = 40
Numero de numeros gerados = 166
Numero de numeros gerados = 41
```

```
Numero de numeros gerados = 29
Numero de numeros gerados = 13
Numero medio de numeros gerados = 487/10 = 48
```

Note que suprimimos, neste programa, a impressão das seqüências geradas.

Experimente executar este programa para vários valores de `n_max` e valores grandes de `num_vezes` e investigue a relação que existe entre o número médio de números gerados e o valor de `n_max`. Naturalmente, quanto maior for `n_max`, maior vai ser esta média. Você pode fazer uns gráficos para tentar entender que tipo de dependência existe entre estas duas quantidades.

5 Valores das partes

Você pode entregar três programas neste exercício, correspondentes aos itens (A), (B), e (C). Os valores destes três programas são como segue:

1. Valor do programa em (A): 3,0 pontos
2. Valor do programa em (B): 7,0 pontos
3. Valor do programa bônus em (C): 1,0 ponto

Boa diversão!