TCC341 - Trabalho 03

Introdução

A criptografia quântica é consiste de criptossistemas que utilizam os princípios da Mecânica Quântica para garantir uma comunicação totalmente segura, isto é, é sempre possível detectar quando um intruso fez qualquer tentativa de fraude. Nela, Alice e Bob conseguem criar e compartilhar uma chave secreta para criptografar e decifrar suas mensagens, normalmente a partir da polarização de fótons.

Objetivo

Nesse trabalho você deve simular uma distribuição de chaves quântica. Seu programa deve simular o lado de Alice, gerando os fótons e imprimindo-os na tela. Ao recebeer a escolha de bases de Bob você deve imprimir na tela a chave final compartilhada.

Para fazer a escolha aleatória dos bits e da base você deverá usar as funções *srand()* e*rand()* encontradas na biblioteca *stdlib.h*. Você não pode utilizar nenhuma outra biblioteca além dela e da *stdio.h*. Você deve utilizar o valor de semente no *srand()* no início do seu programa e escolher todos os bits seguido de todas as bases separadamente. Para isso faça *rand() % 2*, onde 0 representa a primeira base e 1 representa a segunda. Veja os exemplos para que seu programa execute como esperado.

A tabela abaixo indica os caracteres que devem ser usados para representar cada base e cada polarização:

Base	0	1
+	**^**	**>**
x	**/**	****
o	**@**	**G**

Por exemplo, suponha 8 bits e as bases **+** e **x**. Alice vai escolher uma sequência de bits, por exemplo **01101101**. Para essa sequência ela escolhe as bases **01000000**, ou seja,**+x++++++**. Ao polarizar os fótons, segundo a tabela acima, temos: **^ \ > ^ > > ^ >**. Bob, então, publica sua sequência de bases **+xxx+x++**. Alice compara as bases de Bob com as suas e verifica que 5 são iguais e escolhe os bits daquelas 5 posições para formar a chave: **01--1-01**. Finalmente Alice converte **01101** de binário para decimal, imprimindo **13**na tela, que representa a chave final.

Entrada e Saída

Seu programa deve receber 2 inteiros (s e n), representando o número de bits e o valor da semente para o algoritmo aleatório. O número máximo de bits entrado será 100. Na linha de baixo ele deve receber dois caracteres separados por espaço, representando as bases que serão usadas. Com esses dados, ele deve imprimir a sequência de polarização de n fótons (caracteres separados por espaços).

Após a impressão dos fótons, ele deve ler uma sequência de bases escolhidas por Bob e a partir delas calcular a chave e imprimir na tela, em inteiro positivo. A sequência é dada por n caracteres separados por espaços.

Exemplos

Nos exemplos abaixo a cor azul representa a entrada que deve ser lida pelo seu programa e a cor vermelha, a saída que deve ser gerada pelo seu programa na saída padrão.

Exemplo 1:

```
0 8
+ x
^ \ > ^ > > ^ >
+ x x x + x + +
13
```

Exemplo 2:

```
0 8
x o
/ G \ / \ \ / \
x o o o x o x x
13
```

Exemplo 3:

```
42 100
+ o
^ G > G ^ > G @ ^ > ^ G @ G ^ @ ^ ^ ^ @ G @ ^ G G @ G > G G @
@ > > ^ > ^ > > G ^ > @ > G G @ G > ^ @ ^ G > G ^ @ G ^ @ > > >
> G @ G G G G G @ G ^ ^ G > @ > @ ^ > @ @ @ @ ^ ^ > > G @ @
> @ G G > > >
+ o o o + + + + o + + o + o + o + + + o o o + o + o + o + + o + + o o o o
o o + + + + + + o + o + o + o + o + o + o + o o + o + o + o + o + o + + + o
o o o o + o + o + o o o o o + + o o + o o o o o o o o
30822771656085
```