

1. UAb E-fólio B, 2021, Alínea A

O e-fólio B é constituído por 4 alíneas, valendo 1 valor cada, devendo as mesmas serem realizadas sequencialmente, e podendo ser reutilizado código entre alíneas. A cotação total do e-fólio é de 4 valores. Os critérios de correção encontram-se no espaço da UC, sendo que 50% é destinado à funcionalidade, resultante da percentagem de casos de teste corretos. A realização do e-fólio na plataforma HackerRank não dispensa a entrega do relatório no espaço da UC. Deve em qualquer caso submeter terminando a submissão no HackerRank, de modo a poder ser apurado o critério da funcionalidade.

O relatório deve indicar as alíneas realizadas e resultados obtidos, e descrever o código realizado e opções tomadas, o qual não deve ultrapassar as 4 páginas. Se realizou parcialmente uma das alíneas, descreva o que fez e como planeava completar a alínea. Deve colocar o código das alíneas realizadas no anexo, mesmo as que foram realizadas parcialmente, e não colocar código no corpo do relatório.

Neste e-fólio deve alocar memória dinamicamente, dependente da entrada de dados (utilizando malloc e free). Caso pretenda fazer o e-fólio para metade da cotação, pode alocar memória de forma estática, independente da entrada de dados (penalização apenas nas alíneas C e D).

Considere que não existem linhas maiores ou iguais a 1024 caracteres. Deve considerar a entrada de dados pelo stdin.

O enunciado inclui uma história envolvendo um cliente, em continuação do e-fólio A, em que você é contratado(a) para desenvolver um programa. Pretende-se desta forma aproximar o e-fólio de uma situação real.

Enunciado:

Epaminondas suspeita que esteja a ser observado e precisa de poder guardar as suas ideias num ficheiro que apenas ele possa ler. No entanto não confia nos programas de encriptação atuais, mas teve uma ideia para um algoritmo de encriptação e já lhe deu um nome, o INQUEBRÁVEL. Precisa dos seus serviços apenas para implementar o algoritmo, e embora tenha ganho alguma confiança nos seus serviços, devido ao seu trabalho realizado no e-fólio A, Epaminondas não tem a certeza que consiga desta vez implementar exatamente o que pretende, e assim sendo não vai revelar mais que o necessário a cada momento.

A codificação de um ficheiro faz-se através de uma chave, a qual é necessária para descodificar o ficheiro. No entanto em vez de uma string, Epaminondas precisa de converter cada chave num número sem sinal, para realizar uma operação secreta, a revelar no decurso do algoritmo. Pretende-se que leia a chave e devolva o número sem sinal correspondente. O processo deve ser o seguinte:

1. Inicializar o resultado a 0
2. Processar o primeiro caracter, C, fazendo:
3. Somar o código do caracter ao resultado
4. Fazer um XOR binário do resultado com ele próprio após deslocar 26 bits para a esquerda
5. Fazer novo XOR binário do resultado com ele próprio, após deslocar 15 bits para a esquerda
6. Fazer novo XOR binário do resultado com ele próprio, após deslocar 7 bits para a esquerda
7. Se C for o último caracter, parar, caso contrário passar para o próximo C e voltar ao passo 3

No caso da chave ser apenas uma letra, digamos que a letra 'a', os passos seriam os seguintes:

- (passo 1) resultado=0
- (passo 2) C='a' (código 97)
- (passo 3) resultado=97 (soma de 0 com 97, em binário é 0000 0000 0000 0000 0000 0000 0110 0001)
- (passo 4) 0000 0000 0000 0000 0000 0000 0110 0001 XOR 1000 0100 0000 0000 0000 0000 0000 0000
 - resultado: 1000 0100 0000 0000 0000 0000 0110 0001
- (passo 5) 1000 0100 0000 0000 0000 0000 0110 0001 XOR 0000 0000 0011 0000 1000 0000 0000 0000
 - resultado: 1000 0100 0011 0000 1000 0000 0110 0001
- (passo 6) 1000 0100 0011 0000 1000 0000 0110 0001 XOR 0000 0000 0000 0000 0011 0000 1000 0000
 - resultado: 1000 0100 0011 0000 1011 0000 1110 0001
- (passo 7) terminar, já que não há mais caracteres caso contrário seguira para o passo 3 para processar o carácter seguinte. O resultado é 2 217 783 521,

Notar que a verde estão os 0 adicionados devido ao deslocamento dos bits para a esquerda.

Preocupado com que este procedimento não seja bem implementado, Epaminondas construiu a seguinte tabela com as chaves e valores correspondentes:

Entrada	Saída
a	2217783521
bsd4d	70687441
asd4d	406264400
2fvqwEWqfg	1150541008
5	3558513333
321351	731556911
AAA	2822799811
21090	110246140
21173	779287038
Epaminondas	3875100399

```

1  #include <math.h>
2  #include <stdio.h>
3  #include <string.h>
4  #include <stdlib.h>
5  #include <assert.h>
6  #include <limits.h>
7  #include <stdbool.h>
8
9  int main()
10     /* Enter your code here. Read input from STDIN. Print output to STDOUT */
11     return 0;
12

```

2. UAb E-fólio B, 2021, Alínea B

Epaminondas pretende agora que processe um ficheiro de texto. No entanto esse ficheiro irá ter caracteres codificáveis, e outros que não pretende que sejam alterados. A lista de caracteres codificáveis é a seguinte:

```
"ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789abcdefghijklmnopqrstuvwxyz \t"
```

A ordem dos caracteres é importante numa fase posterior, existindo num total 64 caracteres codificáveis. Processe todo o conteúdo dos dados de entrada, e devolva apenas os caracteres codificáveis.

Foi construído para o efeito os seguintes casos de teste:

Entrada	Saída
Epaminondas pretende agora que processe um ficheiro de texto.	Epaminondas pretende agora que processe um ficheiro de texto
No entanto esse ficheiro irá ter caracteres codificáveis,	No entanto esse ficheiro ir ter caracteres codificveis
e outros que não pretende que sejam alterados.	e outros que no pretende que sejam alterados
A lista de caracteres codificáveis é a seguinte:	A lista de caracteres codificveis a seguinte
Epaminondas pretende agora que processe um ficheiro de texto. No entanto esse ficheiro irá ter caracteres codificáveis, e outros que não pretende que sejam alterados. A lista de caracteres codificáveis é a seguinte:	Epaminondas pretende agora que processe um ficheiro de textoNo entanto esse ficheiro ir ter caracteres codificveise outros que no pretende que sejam alteradosA lista de caracteres codificveis a seguinte
Epaminondas pretende agora que processe um ficheiro de texto. No entanto esse ficheiro irá ter caracteres codificáveis, e outros que não pretende que sejam alterados. A lista de caracteres codificáveis é a seguinte: "ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789abcdefghijklmnopqrstuvwxyz \t" A ordem dos caracteres é importante numa fase posterior, existindo num total 64 caracteres codificáveis. Processe todo o conteúdo dos dados de entrada, e indique quantos caracteres codificáveis contém. Foi construído para o efeito os seguintes casos de teste:	Epaminondas pretende agora que processe um ficheiro de texto No entanto esse ficheiro ir ter caracteres codificveis e outros que no pretende que sejam alterados A lista de caracteres codificveis a seguinteABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789abcdefghijklmnopqrstuvwxyz tA ordem dos caracteres importante numa fase posterior existindo num total 64 caracteres codificveisProcesse todo o contedo dos dados de entrada e indique quantos caracteres codificveis contmFoi construído para o efeito os seguintes casos de teste
<pre>1 #include <stdio.h> 2 3 int main() 4 { 5 printf("Ola Mundo!"); 6 }</pre>	<pre>1 include stdioh23 int main4 5 printfOla Mundo6</pre>

<pre> 1 #include <stdio.h> 2 3 int main() 4 { 5 /* declaração de três variáveis inteiras */ 6 int x=3; 7 int y=7; 8 int aux; 9 10 /* trocar o valor de x com y */ 11 aux=x; 12 x=y; 13 y=aux; 14 15 /* mostrar os valores em x e em y */ 16 printf("x: %d, y: %d",x,y); 17 } </pre>	<pre> 1 include stdioh23 int main4 5 declarao de trs variveis inteiras 6 int x37 int y78 int aux910 trocar o valor de x com y 11 auxx12 xy13 yaux1415 mostrar os valores em x e em y 16 printfx d y dxy17 </pre>
<pre> P. L. Instr. Resultado 1 6 int x=3; x=3 2 7 int x=7; x=3, y=7 3 8 int aux; x=3, y=7, aux=? 4 11 aux=x; x=3, y=7, aux=3 5 12 x=y; x=7, y=7, aux=3 6 13 y=aux; x=7, y=3, aux=3 7 16 printf.. x: 7, y: 3 x=7, y=3, aux=3 </pre>	<pre> PLInstr Resultado 1 6int x3 x32 7int x7 x3 y73 8int aux x3 y7 aux411auxx x3 y7 aux3512xy x7 y7 aux3613yaux x7 y3 aux3716printfx 7 y 3x7 y3 aux3 </pre>
<p>soma.c</p> <p> _ Somar os primeiros N números inteiros, sendo N definido pelo utilizador.</p> <p>Notas:</p> <p>Escreva sem acentos neste e nos restantes exercícios, para que o código funcione sempre correctamente mesmo sem mudar o código de página para 1252.</p> <p>Durante o desenvolvimento do programa, imprima resultados parciais, de forma a garantir o que o programa faz, eventualmente comentando esse código na versão final, dado que esse texto poderá fazer sentido apenas para o programador e não para o utilizador.</p>	<p>somac Somar os primeiros N nmeros inteiros sendo N definido pelo utilizador</p> <p>Notas:Escreva sem acentos neste e nos restantes exercicios para que o cdigo funcione sempre correctamente mesmo sem mudar o cdigo de pgina para 1252</p> <p>Durante o desenvolvimento do programa imprima resultados parciais de forma a garantir o que o programa faz eventualmente comentando esse cdigo na verso final dado que esse texto poder fazer sentido apenas para o programador e no para o utilizador</p>

```

1  #include <math.h>
2  #include <stdio.h>
3  #include <string.h>
4  #include <stdlib.h>
5  #include <assert.h>
6  #include <limits.h>
7  #include <stdbool.h>
8
9  int main() {
10     /* Enter your code here. Read input from STDIN. Print output to STDOUT */
11     return 0;
12 }

```

3. UAb E-fóio B, 2021, Alínea C

Nesta fase, quando já tem uma semente aleatória da alínea A, e consegue obter os caracteres codificáveis da alínea B, Epaminondas vai revelar agora o algoritmo INQUEBRÁVEL, que codifica um conjunto de caracteres codificáveis com base numa chave. Considere que tem a chave na primeira linha, e pode utilizar o código da alínea A para obter a semente aleatória, e na segunda linha dos dados de entrada estão todos os caracteres que são codificáveis, os quais se pretende codificar, resultado da alínea B.

O algoritmo pretendido, vai baralhar a posição original das letras, e de seguida trocar o valor das letras, e para tal necessita de uma sequência de números aleatórios.

Epaminondas pretende que implemente o seguinte algoritmo, tendo como entrada a string S com os caracteres codificáveis:

1. Inicializar a semente aleatória com o valor obtido através da chave
2. Obter o número de caracteres N existentes na string S a codificar
3. Construir um índice I através da função baralhar (ver Exercício [baralhar.c](#)), utilizando a função aleatória já inicializada
4. Construir um segundo vetor V com N números aleatórios entre 0 e 63, dado que existem 64 letras codificáveis, continuando a utilizar a função aleatória
5. Construir a nova string C já codificada, da seguinte forma:
 1. O elemento i da string C, o caracter C[i] é encontrado na string S na posição I[i]
 2. Deve ser guardado em C[i] não S[I[i]], mas sim o resultado de avançar V[i] caracteres para a frente
 3. O caracter 'A' se avançar 2 para a frente, ficará o caracter 'C', e assim sucessivamente mantendo a ordem dos caracteres da alínea B. No entanto, no caso de chegar ao fim, ou seja, avançaria para a posição 64, volta para a posição 0.
6. Retornar a nova string C

Caracteres codificáveis e ordem definida na alínea B:

"ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789abcdefghijklmnopqrstuvwxyz **m**nopqrstuvwxyz \t"

O algoritmo pode ser exemplificado com o primeiro caso de teste das alíneas A e B (chave: a, S; Epaminondas pretende agora que processe um ficheiro de texto):

- (passo 1) semente aleatória é 2217783521
- (passo 2) existem 61 caracteres
- (passo 3) o índice I é o seguinte: 3; 51; 44; 56; 15; 30; 11; 18; 34; 4; 24; 8; 32; 58; 46; 16; 43; 25; 7; 26; 49; 27; 38; 45; 31; 37; 54; 42; 53; 40; 9; 0; 57; 17; 50; 52; 10; 6; 39; 2; 13; 59; 29; 19; 23; 22; 47; 5; 28; 21; 48; 12; 41; 1; 55; 14; 35; 20; 36; 33;
- (passo 4) o vetor V é o seguinte: 20; 63; 38; 44; 24; 9; 11; 51; 31; 32; 63; 49; 31; 54; 59; 6; 5; 19; 51; 37; 10; 29; 12; 54; 7; 24; 10; 57; 10; 41; 7; 45; 41; 31; 43; 48; 22; 51; 24; 39; 52; 15; 37; 32; 42; 14; 14; 54; 53; 3; 42; 53; 2; 60; 29; 63; 60; 8; 14; 46;
- (passo 5) o primeiro caracter está na posição 3 sendo 'm' (a negrito nos dados de entrada), e após avançar 20 posições (na lista de caracteres, dá a volta), fica com o valor 'E'
- (passo 5) o segundo caracter está na posição 51 sendo ' ' (espaço), e após avançar 63 posições (dá a volta até ao caracter anterior), fica com o valor 'z'
- ...
- (passo 6) C é "EzSUPHJ0FMqYUjckkta9 Rq2wOIt07hnaQ3XMbWLFbNI2usjdjdWeolUdaGE6"

A função aleatória a utilizar deverá ser a função randaux, a qual utiliza uma variável global "seed" que deve ser mudada no passo 1.

Epaminondas fez a resolução dos 10 casos de teste das alíneas anteriores, para comparar com a execução do código:

1. EzSUPHJ0FMqYUjckkta9 Rq2w0Ito7hnaQ3XMBWLFBNi2usdjdWeoLudaGE6

2. wEm4usZS7j8WtLkx7W831cziZAIY8nZPjG1PHFnZgS8wvzWQ0REWVV

3. RkYvWDGbtalL2gPtL26eIdSGiAE6U2AVHCioDTuqxq

4. oHVxQRY3k 10b3JTFGBzIByw0XuZiJgs7LZJMtFuvGRFe

5. Lg
DqoEpxXniQCGlDmp00CdlpY8IR7j7z0WfbvNhtU6jK4JZUp3ADWBjon8PEHBzWxthi30yYVCpn1xYWz5guypwWBJ45LSa
b50DiH7XTOQHDMwnKTqWzH9Mgb yzjrTnedE3ZA6h1Yg g4rDIm2yjGmI08MblIgwW4H4qFJCDdsE7fJ9wkJ
10zYTA3dGtTcRsjSIUyg

6. rnp32vVuJy 62H4pQLS056t 16YaLd1XLP59ikKN 8mCXF00oYt5K6aIeZJYTASrxq xEW
wtt96tgGP9APYZUYosuIyr4S7zDMZAYR5ASg0QDEE tx
R22dyBgLaiwjHudtxUuWNg8HVLbALLMMD3lHR4q289GVyxjiaESfTSq iewjYXX56LZKWEef0T2Uz5XB4aH
DJWQFs1AL1Knr1Pai6Ep AhWc3oWLAeugSRV92eLIhF2yy 82wXtQPqVsEk sSCoXjsQvLkAhRfEd
u02YiapFwSCewl5vwoeh9P8VX17ono0c830DJYzKReBqm4kHUF04wVgnBI9cpK0r6M5V77f1i8TDFbdT8MIxJxZbsv
9sYIdrfPcux bFpguzRCp4d0mjL4Wktkw42u 5P5zvNNmqX16hd0woCxnPKNIPxS9YIme7EUtdH x NdPohNl
9yj7PMHdR4uggJFHe7HgaJMzS0XVXj5cjupDgLok1clzL6dRGxY08uuv2qS

7. YFK2lBIhwXHD9nw8e gyJeVnWgnppj7bX4be3IfubVmN0d2zA

8. XBZIWesYaqShgf96Vlx5RGIcWBJ58pq20VIitL600r5M4uNDfTR6TUaEmrpKSwJtFdkRbjdnqrR jMKDKUg68
yhr7m9rUdmn0u0HPLJRhbPibydH6EG 3uhCBdI9kSpN 0wbD10w0ofeh9jQHHvs 7MrMb46LTxY
gs0bvFJYf4sSZOANWVeU zZCGprg5xEIRFjLyPKAdSBNi2P7

9. zv5vUvQVLVZIZjsQ9DyHaaKjrEpcDhPPvSTRqBRrd0 Tm7BRzq0By H1 dCwvjjcmWBNxMsCsJfxWBm
8IKm148ZENAMMAcMrdnkHkXIjlyH3SFve2zcQa2hI1z8rfVTl0D3Xylf3JJqrCgPh4xYsTg2Z03aQYc6M6RbEE8hsNR

10. UJp5YGBvronXzvsQgwnUCgmIdXjTHSC8DGq3jwVMsDzACqQER9sQupaWwGmacaW53Ie bxcMM1fyGzx1
1Fk1o38uj3mid9WoMtkyb00wvnpzaQk6qvrCTNKfUtdV44BEAooHJUq4lDF8U3zrW6BYE7lWiBgFC8oTFekbDrki15ZCsEI
fW2yomULR kkj YxCFYtuuIa1XOPjxqar
0zXncfUrKklSPpdwnBcE2xjhb2NeGEBq4DGHcSLNuW9XQkEMwnFJMwfaU
eH2oy3o6X1KIRBUR0w2SVVlsPL0LI6CK5Ya1qi3CGYN221pfjD ekn45rNfByn
jINyEueWxQEjHilfgGulbq8FJS6UM bMX p1LAFUSUNrsGRMLtTdyzRnZSgv VyWAatjTe6z0FieiHrkMiwK87fY
QrrA8Gy2l7ioIu73Km0b3w6zFwrYuA6Akm2SpnoF0tYvJ6

```
1  static long seed = 1;
2  unsigned int randaux()
3  {
4      return(((seed = seed * 214013L + 2531011L) >> 16) & 0x7fff);
5  }
6
7  #include <math.h>
8  #include <stdio.h>
9  #include <string.h>
10 #include <stdlib.h>
11 #include <assert.h>
12 #include <limits.h>
13 #include <stdbool.h>
14
15 int main() {
16     /* Enter your code here. Read input from STDIN. Print output to STDOUT */
17     return 0;
18 }
```

4. UAb E-fóio B, 2021, Alínea D

Epaminondas pretende agora que construa o algoritmo final, codificando e decodificando. Para tal, na primeira linha é dada a chave (sem espaços), seguida de uma das palavras, "codificar" ou qualquer outra palavra, por exemplo, "decodificar". O resto da linha pode ser ignorada. As restantes linhas devem ser codificadas no caso de "codificar" ou decodificadas nos restantes casos. No entanto, Epaminondas pretende que inclua todos os caracteres não codificáveis sem alterações, fazendo as alterações apenas nos caracteres codificáveis.

O grande segredo de Epaminondas é que o algoritmo de decodificação é praticamente igual ao da codificação, mas tem um pequeno detalhe, o índice e o vetor têm de ser invertidos. Para todo o i , $F[i]=i$, devendo ser utilizado F em vez de I . Por outro lado deve-se adicionar aos caracteres em vez de V , o vetor W , sendo $W[i]=64-V[I[i]]$. Notar que o vetor W não só troca o sinal existente em V , com o valor a adicionar, como troca a ordem dos elementos, para que a ordem seja a do vetor F .

O teste que Epaminondas pretende que faça é a codificação e decodificação de todos os casos de teste codificados na alínea C, devendo os textos originais serem iguais aos da alínea B, utilizando as chaves da alínea A. Um texto codificado, se decodificado com a mesma chave, tem de dar exatamente o texto original, incluindo os caracteres não codificáveis.

Após obter o seu programa Epaminondas ficou bastante satisfeito, mas soube que na UC de Segurança e Redes de Computadores este assunto era abordado, tendo ficado na expectativa de aprender mais sobre o assunto.

```
1  static long seed = 1;
2  unsigned int randaux()
3  {
4      return(((seed = seed * 214013L + 2531011L) >> 16) & 0x7fff);
5  }
6
7  #include <math.h>
8  #include <stdio.h>
9  #include <string.h>
10 #include <stdlib.h>
11 #include <assert.h>
12 #include <limits.h>
13 #include <stdbool.h>
14
15 int main() {
16     /* Enter your code here. Read input from STDIN. Print output to STDOUT */
17     return 0;
18 }
```