

Projeto Prático #1 (Estruturas condicionais e de repetição, Funções)

Professora: Leyza/Myriam

Estudante: _____

Questão 1

Construa um programa que produza uma folha de calendário **para um ano completo**, o qual deve ser informado pelo usuário via teclado! O programa a ser desenvolvido deve levar em conta que o mês de fevereiro pode ter 29 dias (ano bissexto). A formatação de cada mês deve seguir exatamente o modelo ilustrado a seguir.

Abril de 2020						
DOM	SEG	TER	QUA	QUI	SEX	SAB
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30		

Pesquisa: para desenvolver o projeto, você precisa descobrir como calcular o dia da semana de um dia particular! Nos comentários do programa, explique em detalhes a técnica utilizada e indique as referências que você consultou.

Questão 2

Faça o teste de mesa para o seguinte código. Além disso, faça um texto explicando detalhadamente o que os operadores & e <<.

```

1  #include <stdio.h>
2
3  #define SIZE (1 << 4)
4  int main()
5  {
6      int x, y, i;
7      for (y = SIZE - 1; y >= 0; y--, putchar('\n')) {
8          for (i = 0; i < y; i++)
9              putchar(' ');
10         for (x = 0; x + y < SIZE; x++)
11             printf((x & y) ? "  " : "* ");
12     }
13     return 0;
14 }
```

Além disso, discuta:

1. O que muda se o início do código for `#define SIZE (1 << 7)`?
2. Existe outra forma de solucionar esse problema?

Questão 3

Anna vive em Algol e Bob vive em Betelgeuse. Uma longa distância os separa, dado que as estrelas estão em diferentes constelações - Perseus e Orion. Eles encontraram uma forma de se comunicar via e-mail, mas perceberam que algumas letras podem ser alteradas durante a transmissão devido à longa distância.

Anna propôs uma forma simples para tentar identificar erros, supondo que as mensagens contém apenas letras (maiúsculas e minúsculas) e espaços, e que linha de comunicação pode corromper no máximo um bit em cada byte.

A proposta é transmitir esses símbolos usando o código ASCII - por exemplo, ao invés de transmitir 'A', é transmitido o valor 65. Anna e Bob observaram que, para a faixa de valores utilizada para as letras, os valores ocupam apenas um byte e que o seu bit mais significativo não precisa ser usado. Por exemplo, o valor 65 em binário é 01000001 - note que o bit mais à esquerda é zero. Eles decidiram explorar este bit para determinar se a letra transmitida foi corrompida ou não.

A proposta consiste em alterar este bit mais significativo de tal forma que a soma dos bits seja sempre par (2, 4, 6 or 8). Seguem alguns exemplos de codificação:

símbolo	código ASCII original	binário original	nro de bits com valor 1	binário codif. para que a soma dos bits seja par	decimal codif.
'A'	65	01000001	2	01000001	65
'B'	66	01000010	2	01000010	66
'C'	67	01000011	3	11000011	195
','	46	00101110	4	00101110	46
' '	32	00100000	1	10100000	160

Por exemplo, considere o caracter 'C': como a codificação do valor original (67) tem um número ímpar de bits com valor 1, a codificação altera o bit mais significativo de 0 para 1. Com isso, o valor 195 é transmitido.

Como a linha de comunicação pode corromper no máximo um bit em cada byte, aqueles cuja soma dos bits seja um número ímpar são considerados corrompidos e devem ser descartados. Por exemplo, ao receber o valor codificado de 103 (que em binário é 1100111), pode-se afirmar que esse símbolo foi corrompido durante a transmissão, por ter um número ímpar de bits.

Sua tarefa possui duas partes (crie um menu de forma a possibilitar que o usuário escolha o que deseja fazer):

A dada a mensagem de entrada: (a) codificar segundo a regra acima e (b) mostrar o resultado. Exemplo:

Mensagem de entrada:

Anna.

Mensagem codificada:

65 238 238 225 46

B dada a mensagem codificada: (a) checar cada letra e (b) mostrar a mensagem decodificada inserindo um * em letras corrompidas.Exemplo:

Mensagem de entrada (codificada):

65 238 236 225 46

Saída:

An*a.

Assuma que a mensagem pode conter apenas letras (maiúsculas e minúsculas) e espaços. O final da mensagem é sempre dada pelo símbolo ',' - para facilitar a implementação, você pode supor que ele jamais será corrompido.

Questão 4

Um cartão é composto de uma sequência de 13 a 16 dígitos decimais. Os dígitos mais significativos (prefixos) identificam qual a companhia que opera o cartão, de acordo com a seguinte tabela:

Operadora	Prefixo	Tamanho (em dígitos)	Algoritmo de Verificação
Mastercard	51 a 55	16	Luhn 10
Visa	4	13 ou 16	Luhn 10
Amex	34 ou 37	15	Luhn 10
Diners	30 ou 36 ou 38	14	Luhn 10
Discover	6011	16	Luhn 10
enRoute	2014 ou 2149	15	Luhn 10
JCB	3	16	Luhn 10
JCB	2131 ou 1800	15	Luhn 10

O algoritmo de Luhn foi desenvolvido por Hans Peter Luhn, da IBM, em 1954. Ele verifica se um número de cartão é bem formado realizando o seguinte teste:

1. Iniciando com o dígito mais à direita, multiplique por 2 (dois) todos os dígitos de posição par, e deixe inalterados os dígitos das posições ímpares. Note que o dígito mais à direita é indexado por 1, ou seja, sua posição sempre é ímpar. Por exemplo, o número 49927398716 seria reescrito como 4(18)9(4)7(6)9(16)7(2)6. Os parênteses ilustram os dígitos que foram dobrados, e são usados apenas para fins didáticos. Sem os parênteses, o número resultante seria 4189476916726.
2. Some todos os dígitos, independente da sua posição. Assim, o número resultante do passo 1 produziria $4+1+8+9+4+7+6+9+1+6+7+2+6 = 70$
3. Se o número final é congruente a zero em módulo 10 (ou, em termos mais simples, se a divisão por dez produzir um resto igual a zero), então o número inicial é válido.

Faça um programa que, dado um número, identifique a operadora e verifique a validade do número. As situações a serem tratadas e que devem ser indicadas por mensagens adequadas são as seguintes:

1. Números que contiverem caracteres não numéricos (caracteres fora do intervalo [0;9]);
2. Números com tamanho incorreto;
3. Números de operadoras desconhecidas (que não precisam ser validados);
4. Números de operadoras conhecidas, mas inválidos;
5. Números de operadoras conhecidas e válidos.

Nos casos (4) e (5), a mensagem deve incluir o nome da operadora. Casos de teste

Número Caso

354683A6c Caracteres inválidos

347598 Número muito curto

51783746281726178 Número muito longo

7145819143526709 Operadora desconhecida

355601236748643 Operadora desconhecida

213078347298518 Operadora desconhecida

5105105105105100 Mastercard, válido

5555555555554443 Mastercard, inválido

4222222222222222 Visa, válido

4567890123456789 Visa, inválido

378282246310005 Amex, válido

38520000023237 Diners, válido

3566002020360606 JCB, inválido