Trabalho de linguagem de programação

1º semestre

1

DEFINIÇÃO DO PROBLEMA

Escreva programa que simule o lançamento de uma moeda. Imprima uma tabela com a relação entre o número\_de\_caras/número\_total\_de\_lançamentos para 1, 10, 20, 50, 100, 1000 e 10000 lançamentos. Explique o resultado. Use a função rand() para gerar “aleatoriamente valores 0 (caras) e um (coroas) (ver livro texto). Use um for() com contagem ate 10000, e um switch() dentro for() para imprimir cada caso.

SOLUÇÃO

Para solucionar o problema foi criado um laço for para o lançamento da moeda, foi adicionada a função rand() para lançamento aleatório da moeda e adição de 1 no valor da moeda para a estrutura case não dar erro, foi adicionado um comando de decisão if() e vários else..if para imprimir os valores quando o lançamento chegar a um determinado número de vezes jogadas. E uma estrutura switch que separa se vai ser somado +1 em cara ou coroa.

VALIDAÇÃO

Para obter a validação o programa deve imprimir uma tabela informando quantas vezes foi jogada a moeda e de em cada caso, quantas vezes caiu coroa e quantas vezes caiu cara

COMENTÁRIOS E CONCLUSÕES

O resultado saiu conforme o esperado.

LISTAGENS

Código comentado

#include<stdio.h>

#include<stdlib.h>

int main()

{

// atibuicao das variaveis

int a=0, b=0, caras=0, coroas=0;

// adicao da estrutura de repeticao do lancamento da moeda

for(a=0; a<=10000; a++)

{

// lancamento aleatorio da moeda e adicao de 1 no valor da moeda para a estrutura case nao dar erro

b= rand() % 2;

b= b+1;

// estrutura para imprimir os valores quando o lancamento chegar a um determindo numero de vezes jogados

if(a==1)

{

printf("1x %d=caras e %d=coroas\n", caras, coroas);

}

else if(a==10)

{

printf("10x %d=caras e %d=coroas\n", caras, coroas);

}

else if(a==20)

{

printf("20x %d=caras e %d=coroas\n", caras, coroas);

}

else if(a==50)

{

printf("50x %d=caras e %d=coroas\n", caras, coroas);

}

else if(a==100)

{

printf("100x %d=caras e %d=coroas\n", caras, coroas);

}

else if(a==1000)

{

printf("1000x %d=caras e %d=coroas\n", caras, coroas);

}

else if(a==10000)

{

printf("10000x %d=caras e %d=coroas\n", caras, coroas);

}

// separa se vai ser somado +1 em cara ou coroa

switch (b)

{

case 1:

caras++;

// sai da estrutura switch

break;

case 2:

coroas++;

break;

}

}

system("pause");

return 0;

}

Resultados

1x 0=caras e 1=coroas

10x 7=caras e 3=coroas

20x 9=caras e 11=coroas

50x 22=caras e 28=coroas

100x 52=caras e 48=coroas

1000x 493=caras e 507=coroas

10000x 4965=caras e 5035=coroas

2

DEFINIÇÃO DO PROBLEMA

Escreva a função double calcula\_raiz (double num); que recebe um número e calcula a raiz quadrada do mesmo através do algoritmo: novoChute=0.5\*(ultimoChute+num/ultimoChute). Efetue esse processo até que o erro seja inferior a 0.00001. Use chute inicial igual a 1.0.

SOLUÇÃO

Ao declarar a função calcula\_raiz, foi usado um for que irá rodar até que o último chute seja diferente da raiz quadrada do número dado pelo usuário e dentro do loop for, a variável ultimoChute foi igualada a variável novoChute, assim o ultimoChute será o novoChute da próxima variável

VALIDAÇÃO

Para ter a validação é necessário que ao executar o programa ele dê a raiz quadrada do número dado pelo usuário.

COMENTÁRIOS E CONCLUSÕES

O programa saiu conforme o esperado.

LISTAGENS

Código comentado

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

// prototipo da funcao

double calcula\_raiz (double num);

int main()

{

int num;

double ultimoChute, novoChute, resultado;

printf("Digite um numero para descobrir sua raiz.\n");

scanf("%d", &num);

resultado = calcula\_raiz (num); //chamada da funçao

printf("Resultado = %lf", resultado);

return 0;

}

//declaracao da funcao

double calcula\_raiz (double num){

double ultimoChute, novoChute;

for(ultimoChute=1; ultimoChute!=sqrt(num);)

{

novoChute=0.5\*(ultimoChute+num/ultimoChute);

ultimoChute= novoChute;

}

num= ultimoChute;

return num;

}

Resultados:

Digite um numero para descobrir sua raiz.

25

Resultado = 5.000000

3

DEFINIÇÃO DO PROBLEMA

Escreva programa para ler uma seqüência de notas de alunos (entre 0.0 e 10.0), de tamanho desconhecido, terminadas com um valor negativo. Seu programa não deve armazenar as notas individuais mas apenas fornecer como saída a média da turma, a menor nota e a maior nota.

SOLUÇÃO

Foi criado um laço while() para fornecer um tamanho desconhecido e terminar ao ser enviado uma nota de negativa, foi adicionado dentro do loop a expressão snotas = snotas + nota, assim pode-se somar as notas fornecidas e no final para calcular a média foi dividida a variável snotas pela variável cont, a qual conta quantas vezes foi rodado o loop(número de alunos).

VALIDAÇÃO

Para obter a validação ao receber os dados o programa deve mostrar a média, a menor e a maior nota.

COMENTÁRIOS E CONCLUSÕES

Os resultados saíram conforme o esperado

LISTAGENS

Código comentado

#include<stdio.h>

#include<stdlib.h>

#include<math.h>

int main()

{

// atribuicao de variaveis

float nota, media, cont=0, snotas=0, maior=0, menor=10;

printf("digite as notas dos alunos, e para concluir digite um numero negativo\n");

// loop para adicionar nota ate a adicao de um valor negativo

while(nota>=0)

{

scanf("%f", &nota);

// estrutura de decisao para saber se a nota esta entre 0 e 10

if(nota>10)

{

printf("Digite um numero entra 0 e 10\n");

}

else if(nota<0)

{

break;

}

else

{

cont++;

snotas= snotas+nota;

// estrutura para substituir a menor nota por uma nova menor nota

if(nota>maior)

{

maior=nota;

}

// estrutura para substituir a maior nota por uma nova maior nota

if(nota<menor)

{

menor=nota;

}

}

}

media = snotas/cont;

// impressao das notas

printf("A menor nota = %.2f\n", menor);

printf("A maior nota = %.2f\n", maior);

printf("Sua media = %.2f", media);

return 0;

}

Resultados

digite as notas dos alunos, e para concluir digite um numero negativo

1

2

3

4

5

6

-1

A menor nota = 1.00

A maior nota = 6.00

Sua media = 3.50

4

DEFINIÇÃO DO PROBLEMA

Calcule o fatorial de um dado número inteiro, N, inserido pelo usuário. Seu programa deve fornecer como saída uma tabela com duas colunas, n e o seu fatorial, com n variando de 1 até N. Observe a tabela e descubra qual o máximo valor de N inteiro (NMAX) cujo fatorial pode ser calculado usando int. Como se pode obter fatorial para valores maiores que NMAX?

SOLUÇÃO

É pedido que o usuário digite um número por scanf(), depois é usada a estrutura de repetição, for, para calcular o fatorial e imprimir uma tabela com os valores , a estrutura de decisão, if, para deixar o resultado positivo caso esteja negativo e gotoxy() com printf() no corpo para mostrar onde o valor deve ser imprimido na tabela.

VALIDAÇÃO

Para obter a validação o programa deve fornecer uma tabela com duas colunas correspondendo o número ao seu fatorial

COMENTÁRIOS E CONCLUSÕES

O código teve algumas discrepâncias, pois o fatorial de 32 e 33 sai negativo mesmo após uma estrutura de decisão com inversão de valor caso negativo

LISTAGENS

Código comentado

#include <stdio.h>

#include <stdlib.h>

#include <conio.h>

#include <windows.h>

// funcao para adicao de coordenadas

int gotoxy( const WORD x, const WORD y )

{

COORD xy;

xy.X = x;

xy.Y = y;

return SetConsoleCursorPosition( GetStdHandle( STD\_OUTPUT\_HANDLE ), xy );

}

// funcao principal

int main()

{

// atribuicao de variaveis

int a, b, c=1, d=1;

// pede que o usuario digite um numero

printf("Digite um numero entre 0 e 33 para receber seu fatorial.\n");

scanf("%d", &a);

// estrutura de repeticao para calcular o fatorial e imprimir uma tabela com os valores

for(b=2; b<=a; b++)

{

c= c\*b;

d++;

// estrutura de decisao para deixar o resultado positivo caso esteja negativo

if(c<0)

{

c= -c;

// mostra ande o valor deve ser imprimido na tabela

gotoxy(1,d);

{

printf("Numero %d", b);

}

gotoxy(12, d);

{

printf("Fatorial %d",c);

}

}

else

{

gotoxy(1,d);

{

printf("Numero %d", b);

}

gotoxy(12, d);

{

printf("Fatorial %d",c);

}

}

}

// impressao do resultado da fatoracao

printf("\n %d foi o numero fatorado e %d foi o resultado da fatoracao.\n", a, c);

system("pause");

return 0;

}

//16! foi o maximo sem erro em inteiro e 33 foi o maximo retornando um valor

// para aumentar o NMAX deve ser trocado %d por %f ou %lf

Resultados

Digite um numero entre 0 e 33 para receber seu fatorial.

33

Numero 2 Fatorial 2

Numero 3 Fatorial 6

Numero 4 Fatorial 24

Numero 5 Fatorial 120

Numero 6 Fatorial 720

Numero 7 Fatorial 5040

Numero 8 Fatorial 40320

Numero 9 Fatorial 362880

Numero 10 Fatorial 3628800

Numero 11 Fatorial 39916800

Numero 12 Fatorial 479001600

Numero 13 Fatorial 1932053504

Numero 14 Fatorial 1278945280

Numero 15 Fatorial 2004310016

Numero 16 Fatorial 2004189184

Numero 17 Fatorial 288522240

Numero 18 Fatorial 898433024

Numero 19 Fatorial 109641728

Numero 20 Fatorial 2102132736

Numero 21 Fatorial 1195114496

Numero 22 Fatorial 522715136

Numero 23 Fatorial 862453760

Numero 24 Fatorial 775946240

Numero 25 Fatorial 2076180480

Numero 26 Fatorial 1853882368

Numero 27 Fatorial 1484783616

Numero 28 Fatorial 1375731712

Numero 29 Fatorial 1241513984

Numero 30 Fatorial 1409286144

Numero 31 Fatorial 738197504

Numero 32 Fatorial -2147483648

Numero 33 Fatorial -2147483648

33 foi o numero fatorado e -2147483648 foi o resultado da fatoracao.

Pressione qualquer tecla para continuar. . .

5

DEFINIÇÃO DO PROBLEMA

Escreva uma função receba um inteiro positivo e retorne 0 (zero) se o número for primo ou 1 (um) se ele não for primo.

SOLUÇÃO

Para solucionar, primeiramente, foi usado scanf() para o usuário informar um número inteiro qualquer, em seguida foi usada a função inteiro e dentro dela um comando if, para caso o número informado for igual a 2, retornar 1. Após foi criado um laço for com um comando if, contendo uma expressão lógica que retorna 0 se o resto da divisão for igual a zero, pois se o número for divisível por outro que não seja 1 e ele mesmo, não será primo.

VALIDAÇÃO

Para obter a validação, após digitar um número, o programa deve informar se o número é primo ou não.

COMENTÁRIOS E CONCLUSÕES

O programa saiu como o esperado no início.

LISTAGENS

Código comentado

#include<stdio.h>

#include<stdlib.h>

#include<conio.h>

int inteiro(int n1); //Prototipo da funcao

int main(){

int num1, valor;

printf("Digite um numero inteiro positivo\n");

scanf("%d", &num1);

valor = inteiro(num1); //Chamada da funcao

if(valor == 1){

printf("Numero eh primo");

}

else{

printf("Numero nao eh primo");

}

return 0;

}

int inteiro(int n){

int n1;

if(n==2)

return 1;

for(n1=2; n1<=n/2; n1++) // Laço for

{

if(n % n1==0){

return 0;

}

}

return 1;

}

Resultados

Digite um numero

55

Numero nao e primo

Digite um numero

7

Numero e primo

6

DEFINIÇÃO DO PROBLEMA

Escreva programa para determinar se um número inteiro lido via teclado é palíndromo

SOLUÇÃO

Primeiro é pedido que o usuário digite um número via scanf(), em seguida, foi usado um laço while() com duas expressões no seu corpo, assim o número dado será invertido e por último, um comando de decisão if() que informa se é palíndromo ou não.

VALIDAÇÃO

Para obter a validação, após digitar um número qualquer, o programa deve informar se é palíndromo ou não.

COMENTÁRIOS E CONCLUSÕES

O resultado saiu conforme o esperado.

LISTAGENS

Código comentado

#include<stdio.h>

#include<stdlib.h>

#include<math.h>

int main()

{

// atribuicao de variaveis

int a, b, c=0, d;

// pede para o usuario digitar um numero

printf("Digite um numero para descobrir se ele e palindromo.\n");

scanf("%d", &a);

b=a;

// estrutura de repeticao para inverter o numero dado

while(b!=0)

{

c= c\*10+b%10;

b= b/10;

}

// estrutura de decisao para ver se é palindromo

if(c==a)

{

printf("%d e palindromo.\n", a);

}

else

{

printf("%d nao e palindromo.\n", a);

}

system("pause");

return 0;

}

Resultados

Digite um numero para descobrir se ele e palindromo.

123456789

123456789 nao e palindromo.

Digite um numero para descobrir se ele e palindromo.

1234321

1234321 e palindromo.

7

DEFINIÇÃO DO PROBLEMA

Implemente e teste programa para solucionar o jogo torres de hanói.

SOLUÇÃO

Em primeiro, foi atribuído valores nas variáveis do void mover(int, char, char, char), ao definir a função, foi criado um comando de decisão if...else que decide que tipo de movimento qual disco irá fazer, depois imprime o movimento que fez.

VALIDAÇÃO

Para obter a validação o programa deve imprimir as instruções para completar a torre de hanói.

COMENTÁRIOS E CONCLUSÕES

O resultado saiu conforme o esperado.

LISTAGENS

Código comentado

#include<stdio.h>

#include<stdlib.h>

void mover(int, char, char, char);

int main()

{

// atribuicao de valores nas variaveis do void mover(int, char, char, char)

mover(3,'0', 't', 'd');

system("pause");

return 0;

}

// funcao com atribuiçao de variaveis

void mover(int n, char Orig, char Temp, char Dest)

{

// decide que tipo de movimento qual disco ira fazer, depois imprime o movimento que fez

if(n==1)

{

printf("Mova o disco 1 da haste %c para a haste %c\n", Orig, Dest);

}

else

{

mover(n-1, Orig, Dest, Temp);

printf("Mova o disco %d da haste %c para a haste %c\n", n, Orig, Dest);

mover(n-1, Temp, Orig, Dest);

}

}

Resultados

Mova o disco 1 da haste 0 para a haste d

Mova o disco 2 da haste 0 para a haste t

Mova o disco 1 da haste d para a haste t

Mova o disco 3 da haste 0 para a haste d

Mova o disco 1 da haste t para a haste 0

Mova o disco 2 da haste t para a haste d

Mova o disco 1 da haste 0 para a haste d

8

DEFINIÇÃO DO PROBLEMA

Considere uma escada com n degraus e, que a cada instante você consegue, com um passo, subir 1 ou 2 degraus. Quantas são as formas diferentes de subir os degraus da escada? Apresente duas soluções, uma recursiva e outra não-recursiva.

SOLUÇÃO

Na solução não recursiva, é pedido que o usuário digite o número de degraus de uma escada que pode ser subida utilizando 1 ou 2 passos, após foi criado um laço for(), o qual possui em seu corpo uma expressão que irá, de acordo com o loop, calcular as maneiras de subir a escada de forma não recursiva.

VALIDAÇÃO

Para obter a validação o programa deve informar quantas maneiras pode-se subir a escada de acordo com o número passado pelo usuário.

COMENTÁRIOS E CONCLUSÕES

A solução não recursiva saiu de acordo com o esperado e não da para executar de forma recursiva, pois há um limite de passos que possa dar ao mesmo tempo.

LISTAGENS

Código comentado

#include<stdio.h>

#include<stdlib.h>

#include<math.h>

int main()

{

float a, b, c; //declaração das variáveis

printf("Digite o numero de degraus de uma escada que pode ser subida utilizando 1 ou 2 passos.\n");

scanf("%f", &a); //usuário digita um número

for(b=a; b>=1; b--) // laço for com uma expressão que calcula quantas maneiras possíveis de subir

c= c+b/2;

printf("%.0f maneiras de subir a escada de forma nao recursiva.\n", c); // impressão

system("pause");

return 0;

}

Resultados

Digite o numero de degraus de uma escada que pode ser subida utilizando 1 ou 2 passos.

22

127 maneiras de subir a escada de forma nao recursiva.

9 -

DEFINIÇÃO DO PROBLEMA

Escreva programa para demonstrar escopo de variáveis, global, local, static.

SOLUÇÃO

Para solucionar foi criado uma variável global, local e estatic e de acordo com o programa foi se mostrando como elas são utilizadas corretamente. A global localizada antes de main, a local está no corpo de main e a static associada à uma função.

VALIDAÇÃO

Para obter a validação o programa de executar corretamente.

COMENTÁRIOS E CONCLUSÕES

Não houve discrepâncias e o resultado saiu conforme o esperado

LISTAGENS

Código comentado

#include<stdio.h>

#include<stdlib.h>

int num = 20; //num é uma varável global, pois ele pode ser acessada por qualquer função

int multiplica(int resultado);

int main(){

int n1 = 10, j, r; //n1, j e r são uma variáveis locais, pois elas podem ser vistas e manipuladas dentro da função main

printf("n1 equivale = %d\n", n1);

r = multiplica(j);

printf("o resultado = %d", r);

return 0;

}

static int multi = 2; //static pode ser utilizado para variáveis globais e locais, e tem como função indicar se a variável é permanente

multiplica(int resultado)

{

resultado = num \* multi;

return resultado;

}

Resultados

n1 equivale = 10

o resultado = 40