

Conceitos Básicos de Programação

```
1 #include <iostream>
2 #include <vector>
3 #include <algorithm>
4
5 using namespace std;
6
7 int number = 1;
8
9 void fill_vector(vector<int> &v, int (*f)(void)) {
10     generate(v.begin(), v.end(), (*f));
11 }
12
13 int fill_balls() {
14     int in;
15     cin >> in;
16     return in;
17 }
18
19 int fill_combinations() {
20     return number++;
21 }
22
23 bool is_possible(vector<int> balls, vector<int> combinations) {
24     for (int i = 0; i < balls.size() - 1; i++) {
25         for (int j = i + 1; j < balls.size(); j++) {
26             int value = abs(balls.at(i) - balls.at(j));
27             int position = find(combinations.begin(), combinations.end(), value) - combinations.begin();
28             if (combinations.size() > position) {
29                 combinations.erase(combinations.begin() + position);
30                 if (combinations.size() == 0) {
31                     return true;
32                 }
33             }
34         }
35     }
36     return combinations.size() == 0;
37 }
38
39 int main(int argc, char **argv) {
40     int n = 1, b = 1;
41     while (cin >> n >> b && n != 0 && b != 0) {
42         vector<int> combinations(n), balls(b);
43         fill_vector(combinations, fill_combinations);
44         fill_vector(balls, fill_balls);
45         cout << (is_possible(balls, combinations) ? "Y" : "N") << "\n";
46         number = 1;
47     }
48     return 0;
49 }
```



Lógica

- Forma disciplinada de raciocinar
- Encadeamento coerente de proposições ou ideias
- Maneira pela qual instruções, assertivas e pressupostos são organizados num algoritmo para viabilizar a implantação de um programa (DICIONÁRIO MICHAELIS, 2018).



Algoritmo

Conjunto de regras, operações e procedimentos, bem definido e logicamente ordenado (sequenciado), destinado a resolver um problema ou uma classe de problemas, em um número finito de etapas.



Pseudocódigo e Linguagem de Programação

- **Pseudocódigo: linguagem natural** para descrever uma sequência de ações a fim de se alcançar um determinado objetivo.
- **Linguagem de programação: linguagem formal**, utilizada para padronizar instruções inteligíveis a um computador.



Compilação e Interpretação

- **Interpretação:** verifica uma linha ou comando de um programa, transforma-o em código compreensível ao computador e então realiza sua execução. O procedimento é então repetido, até que as instruções sejam finalizadas, ou seja, **verifica o código linha a linha**.
- **Compilação:** **verifica todo o código**, normalmente, da esquerda para a direita e de cima para baixo, então transforma-o em linguagem de máquina (inteligível ao computador).



C e C++

C++ é uma versão estendida e melhorada da linguagem de programação C, uma vez que suporta todos os seus recursos e, adicionalmente, o paradigma de Programação Orientada a Objetos. Sendo assim, C++ é um superconjunto da linguagem C.

Por isso: é necessário compreender os fundamentos da linguagem C para desenvolver sistemas em C++.



Um programa em C

```
#include <stdio.h>

/**
 * Imprime os números primos de n até m.
 */
int main(int argc, char** argv) {
    int n, m, contador;
    printf("Informe 2 valores inteiros e positivos:\n");
    scanf("%i %i", &n, &m);
    for (int i = (n < 2 ? 2 : n); i <= m; i++) {
        contador = 0;
        for (int j = 2; j < i; j++) {
            if (i % j == 0) {
                contador++;
            }
        }
        if (contador == 0) {
            printf("%i ", i);
        }
    }
    printf("\n");
    return 0;
}
```



Operadores Aritméticos

Precedência:

1ª

- Parênteses: $y / (4 + 2)$

2ª

- Pré-decremento: $++x$; Pós-incremento: $x++$; Pré-decremento: $--x$; Pós-decremento: $x--$

3ª

- Exponenciação: x^n
- Radiciação: \sqrt{x}

4ª

- Multiplicação: $y * y$
- Divisão: x / y
- Módulo da divisão (resto): $x \% y$

5ª

- Soma: $x + y$;
- Subtração: $x - y$;



Operadores Relacionais

Precedência:

1ª

- Maior que: >
- Maior que ou igual: >=
- Menor que: <
- Menor que ou igual: <=

2ª

- Igual: = (==)
- Diferente: ≠ (!=)



Operadores Lógicos

Precedência:

1ª

- Não: !

2ª

- E: &&

3ª

- Ou: ||
- Ou Exclusivo: ^



Operadores Bit a Bit

Precedência:

- E: &
- Ou: |
- Ou Exclusivo: ^
- Complemento de um: ~
- Deslocamento para esquerda: <<
- Deslocamento para direita: >>



Tipos Básicos (ANSI)

- char: 'c', -127, 255
- int: -32767, 65535
- Float: -0.32, 3.1415
- double: 3.1415926
- void



Modificadores

- signed
- unsigned
- short
- long



Estruturas condicionais

- `if () { }`
- `if () { } else { }`
- `if () { } else if () { } else { }`
- `switch () { case ... }`



Estruturas de repetição

- while () {}
- do {} while ()
- for (start; condition; increment) {}



Outro programa em C

```
/*
 * Autor:
 * Data: 11/08/2018.
 * Descrição :Simula a temperatura de arbetura de um chuveiro
 * automático (splinker) tipo amploa de vidro conforme ABNT NBR 6135:1992.
 * Versão: 0.01.
 */
#include <stdio.h>
#include <string.h>
#include <time.h>
#include <unistd.h>
#include <stdlib.h>

int main(int argc, char **argv) {
    unsigned int temperature, is_active = 0;
    const char COLORS[9][9] = {
        "incolor",
        "laranja",
        "vermelho",
        "amarelo",
        "verde",
        "azul",
        "roxo",
        "preto",
        "inválida"
    };
    char color[9];
    for (int index = 0; index < 100; index++) {
        temperature = time(NULL) % 2 == 0 ? time(NULL) % 261 + 32 : rand() % 261 + 20;
        sleep(temperature % 2 + 1);
        is_active = temperature < 57 ? 0 : 1;
        if (temperature < 57) {
            strcpy(color, COLORS[0]);
        } else if (temperature < 68) {
            strcpy(color, COLORS[1]);
        } else if (temperature < 79) {
            strcpy(color, COLORS[2]);
        } else if (temperature < 93) {
            strcpy(color, COLORS[3]);
        } else if (temperature < 141) {
            strcpy(color, COLORS[4]);
        } else if (temperature < 182) {
            strcpy(color, COLORS[5]);
        } else if (temperature < 260) {
            strcpy(color, COLORS[7]);
        } else {
            strcpy(color, COLORS[8]);
        }
        printf("Temperatura nominal: %hu. Coloração líquido: %s.\n", temperature, color);
    }
    return 0;
}
```



Referências

- SCHILDT, H. C: **Completo e Total**. 3 ed. São Paulo: Makron Books, 1996. 720 p.
- DEITEL, H. M.; DEITEL, P. J. Como Programar em C. 6 ed. São Paulo: Pearson, 2011. 818 p.
- DAMAS, L. Linguagem C. 10 ed. Rio de Janeiro: LTC 2007. 410 p.



Dúvidas



Vinícius Silva Madureira Pereira

bit.do/plcs110818

Obrigado...

Bom
fim de semana
para
todos!!!

