Universidade Estadual de Maringá

Centro de Tecnologia - Departamento de informática

Curso de Bacharelado em Informática

Projeto e Análise de Algoritmos

Profº. Dr. Rodrigo Calvo

Acadêmicos	**RA**
Luiz Flávio Pereira	91706
Marcos Vinícius Peres	94594

Maringá - PR

2018

Configurações do computador:

Para o desenvolvimento deste programa, foi utilizado o sistema operacional Windows 10, 64 bits Pt-br. O programa foi desenvolvido na linguagem de programação C, usando a Ide DEV C++, tendo como o compilador TDM-GCC 4.9.2 64 bits Release.

Relatório do programa:

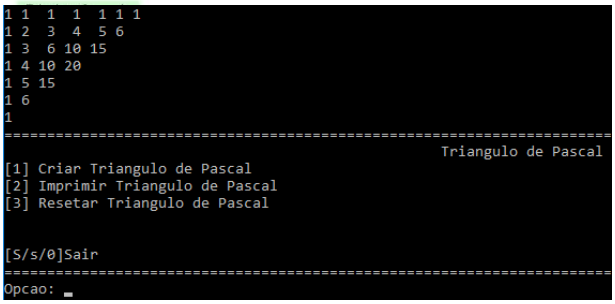
O programa main consiste em um *switch case*, com 3 casos possíveis:

1 – Criar o Triângulo de Pascal

2 – Imprimir o Triângulo de Pascal

3 – Resetar Triângulo de Pascal

0/S/s – Sair do Programa



O programa cria um triângulo de pascal com a aparência da Figura 1.

Figura 1 : Interface do Programa Dinâmico Triângulo de Pascal

A função **cria_trPascal(int matriz[][])** possui duas estruturas de repetição aninhadas para poder percorrer toda a matriz preenchendo-a com seus devidos valores. A borda é preenchida com 1 quando a coluna = 0 ou quando a linha = 0. O interior do triângulo de pascal é preenchida a partir através do seguinte cálculo:

matriz[linha][coluna] = (matriz[linha][coluna - 1]) + (matriz[linha - 1][coluna]);

Essa linha de código em particular é quem caracteriza a programação dinâmica, pois para obter um determinado valor, ela necessita de somar 2 valores menores já conhecidos e salvos em outro endereço da matriz em questão.

A função **reseta_trPascal(int matriz[][])** percorre toda a matriz com o uso de 2 laços aninhados, inserindo o valor 0 a todas as posições da matriz.

A função **recebe_tamanhosUltimaColuna(int matriz[][], int vetor[])** percorre todas as linhas da matriz procurando o maior valor de cada linha e armazenando no vetor a quantidade de digitos que este maior valor possui. Para saber a quantidade de dígitos foi usado uma estrutura de *if's* e *else's* para tratar tal condição.

A função **corrige_Formatacao(int vetor[], int indice, int valor)** faz a subtração entre a quantidade de dígitos do maior valor da linha em particular e a quantidade de dígitos do valor atual a ser impresso e em seguida imprime espaços vazios antes do valor do triângulo de pascal.

A função **imprime_trPascal(int matriz[][])** inicia-se verificando se a primeira posição da matriz possui o valor 0, pois se este valor for encontrado então é disparado um gatilho/mensagem de erro dizendo que o triângulo de pascal não foi gerado. Caso o triângulo de pascal já venha a existir, então é declarado um vetor onde o mesmo será preenchido com a quantidade de dígitos do maior valor de cada linha da matriz na função: **recebe_tamanhosUltimaColuna(matriz[][], vetor[])**

Após ter os valores armazenados no vetor, então a matriz começará a ser percorrida fazendo a impressão dos números diferentes de zero e chamando a função **corrige_Formatacao(int vetor[], int coluna, int valor)** para imprimir o valor do triângulo de pascal de maneira organizada e sem os zeros.

A função **showmenu(char opcao)** realizará a interação com o usuário apresentando as opções e suas devidas funcionalidades.

Lista de tratamentos de exceções do programa:

- Se a matriz do triângulo de pascal possui na posição [0][0] o valor zero, então o triângulo de pascal não existe;
- Se no menu inicial for inserido um valor menor que zero ou maior que 3, então o programa apresenta uma mensagem de opção inválida;
- Após o triângulo de pascal ser resetado sua impressão não será realizada por estar completamente preenchido com valores zerados.

Análise de Complexidade do Programa: *cria_trPascal(int matriz[][])*

Na linha 8 temos a seguinte complexidade:

$$\sum_{i=1}^{n-2} i = \frac{n(n+1)}{2} - (n-1) - n \approx \frac{n^2}{2}$$

Para essa linha de código em especial ignoramos alguns cálculos e arredondamos a complexidade da linha em questão para $\approx \frac{n^2}{2}$.

Código	Compl.

```c
void cria_trPascal(int trPascal[TAMANHO][TAMANHO]){
    for(int linha = 0; linha < TAMANHO; linha++){
        for(int coluna = 0; coluna < TAMANHO; coluna++){
            if(coluna == 0 or linha == 0){
                trPascal[linha][coluna] = 1;
            }else{
                if(coluna < (TAMANHO - linha)){
                    trPascal[linha][coluna] = (trPascal[linha][coluna - 1]) + (trPascal[linha - 1][coluna]);
                }
            }
        }
    }
    printf("Triangulo de Pascal criado com sucesso!\n");
}
```

n_2
n_2
n^2

$n^2 - 2n$
$\frac{n^2}{2}$

1

Deste modo temos:

$$= n + n^2 + n^2 + (n^2 - 2n) + \left(\frac{n^2}{2}\right) + 1$$

$$= 2n^2 + (n^2 - 2n) + \left(\frac{n^2}{2}\right) + n + 1$$

$$= 3n^2 + \left(\frac{n^2}{2}\right) - n + 1$$

$$= O\,(n^2)$$