

UNIVERSIDADE FEDERAL DE VIÇOSA
CAMPUS DE FLORESTAL
CIÊNCIA DA COMPUTAÇÃO

DANIEL FERNANDES PINHO (2634)



FLORESTAL, MG
1 de dezembro de 2021

Sumário

1	Introdução	2
2	Desenvolvimento	3
2.1	Métodos e funções	3
2.2	Menu Inicial	3
2.3	Restrições	4
2.4	Opção 5 - Figura gerada por mim	5
3	Conclusão	6



1 Introdução

Este trabalho tem como objetivo construir quatro figuras compostas integralmente de caracteres do tipo '*' (asterisco) em linguagem de programação e exibi-las em uma tela padrão (terminal) de tamanho 80x20. As figuras são I) Um único asterisco, II) Símbolo de soma (+), III) Letra X e IV) Cruz. O usuário escolhe, através de um menu, qual figura a ser exibida e sua quantidade, entre 1 a 100. Se o usuário digitar um número maior do que 100, será considerado o número 100.



2 Desenvolvimento

Nesta seção, abordarei as principais implementações e os objetivos que me levaram a estas.

2.1 Métodos e funções

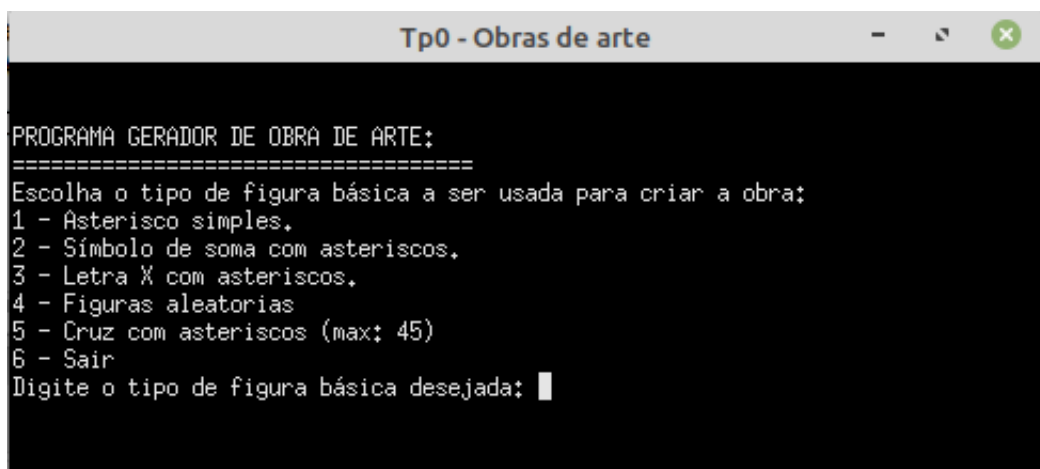
Para utilizar a moldura no programa, foi criado um TAD, que referência aos quadros que serão criados. Segue a restrição de 20x80 (linhas x colunas).

```
1 ~ typedef struct{
2     char quadro[20][80];
3 }TipoQuadro;
4
5 void Inicializa(TipoQuadro *Quad);
6 void ImprimeArt(TipoQuadro *Quad);
7 int SubMenu();
8 void Inicializa(TipoQuadro *Quad);
9 void AsteriscoArt(TipoQuadro *Quad, int qtd,short flag);
10 void MaisArt(TipoQuadro *Quad, int qtd,short flag);
11 void XArt(TipoQuadro *Quad, int qtd,short flag);
12 void AleatorioArt(TipoQuadro *Quad, int qtd,short flag);
13 void CruzArt(TipoQuadro *Quad, int qtd,short flag);
```

Figura 1 – Fonte própria

2.2 Menu Inicial

O menu possui opções simples e claras, conforme segue a imagem de saída:



```
Tp0 - Obras de arte
PROGRAMA GERADOR DE OBRA DE ARTE:
=====
Escolha o tipo de figura básica a ser usada para criar a obra:
1 - Asterisco simples.
2 - Símbolo de soma com asteriscos.
3 - Letra X com asteriscos.
4 - Figuras aleatorias
5 - Cruz com asteriscos (max: 45)
6 - Sair
Digite o tipo de figura básica desejada: █
```

Figura 2 – Fonte própria

Algumas observações são importantes: a opção 4 gera figuras aleatórias incluindo a figura criada por mim. A opção 5 gera as figuras em forma de Cruz, no máximo 45 figuras. Caso digitar um valor menor igual a 0, a quantidade será aleatória.

2.3 Restrições

Para cada tipo de arte citada acima (I, II, III, IV) temos restrições diferentes ao se preencher a matriz. Cada figura possui suas definições e tamanho, então simples testes de posicionamento dos asteriscos garantem que não ocorra sobreposição. Sendo assim, primeiro são gerados dois valores, i e j, que designam a posição central da futura figura. Caso esta posição esteja vazia, pode-se então testar as demais posições, para então preencher as necessárias. Para testar e preencher essas posições, um simples laço de repetição é suficiente.

```
1 void AsteriscoArt(TipoQuadro *Quad, int qtd, short flag){
2
3     int i,j,k, opcao=0;
4     srand(time(NULL));
5
6     if (qtd >100)
7         qtd = 100;
8
9     else if (qtd <= 0) qtd = rand() % 100; //para valores
        menores iguais a 0, deve-se gerar uma quantidade
        aleatória
10
11     for (k = 0; k < qtd; k++){
12         i = 1 + rand() % 18; // para i [1,19]
13         j = 1 + rand() % 77; // para j [1,78]
14
15         if (Quad->quadro[i][j] != '*') //trata caracteres ja
            existentes
16             Quad->quadro[i][j] = '*';
17
18         else
19             k--; //valor decrementado, pois houve conflito de
                valores.
20     }
21
22
23     if (flag){
24         ImprimeArt(Quad);
25         Inicializa(Quad);
26
27         printf("Deseja fazer um novo quadro com este mesmo valor? 1
            - sim 2 - nao\n");
28         scanf("%d",&opcao);
29         if (opcao == 1)
30             AsteriscoArt(Quad,qtd,1);
31         else
32             return;
33     }
34 }
```

Figura 3 – Fonte própria

Foi utilizado as funções **srand(time(NULL))**, e também a **rand()** mod N, sendo que

a última fornece valores aleatórios, com N o valor máximo. Observe o seguinte:

```
i = 1 + rand() % 18; // para i [1,19]
j = 1 + rand() % 77; // para j [1,78]
```

Figura 4 – Fonte própria

Os valores, mesmo sendo aleatórios, devem pertencer a certo intervalo de valores. Assim, elimina-se a possibilidade de coincidir em uma das bordas do quadro (aqui representadas como “|” e “-”).

2.4 Opção 5 - Figura gerada por mim

Criei uma figura maior que as demais, uma **Cruz** formada com asteriscos. A forma de interação é semelhante às demais: geram-se valores para i e j, verifica as posições e escreve onde houver espaço. Por ser uma figura maior, houve a necessidade de redução do número máximo de figuras para 45.



3 Conclusão

Com este trabalho foi possível praticar a linguagem C, que será útil para os próximos trabalhos da disciplina. E também explorar formas de lógica de programação. Foi atingido o objetivo do trabalho proposto.

