

Estruturas de controle e repetição

Estruturas de repetição

Taciana de Oliveira Alves | RA 2025030283

Atividade

Atividade prática realizada com a ferramenta GDBOnline para construir o algoritmo em linguagem de programação C.

Após ter escrito a estrutura básica: `#include <stdio.h>` com a função `int main() {` para que o programa seja executado, juntamente com a função `return 0;`} que indica o fim da execução da função; declarei a variável `num (número)` do tipo `int` (inteiro), por se tratar de números inteiros, e a variável `soma` do tipo `int` (inteiro) que recebe o valor `0` para iniciar e acumular os valores digitados pelo usuário.

A estrutura de repetição `while` executa o bloco de repetições entre chaves `{ }` enquanto a condição dentro dele for verdadeira. No caso, a condição é: `num != 0`. A variável `num` (número) com o sinal `!=` que significa “diferente de”, vai se repetir enquanto ela for diferente de `0` ou seja, enquanto o número digitado não for 0.

Em seguida, pedimos ao usuário para digitar um número. A função `printf` exibe a mensagem pedindo ao usuário para digitar um número ou 0 para encerrar a tarefa: “**Digite um número ou 0 para sair:**”. A sintaxe termina com ponto e vírgula para indicar o fim do comando, e as aspas duplas indicam que o que está dentro delas é um texto.

A função `scanf` lê os dados digitados pelo usuário e armazena na variável `num`. O especificador `%d` indica que o número inteiro lido será armazenado na variável `num`, e o operador `&` indica o endereço da variável `num`.

A condicional `if (num == 0)`, foi colocada para que assim que o usuário digitar o número 0, o laço de repetição while seja encerrado. O comando `break` checa com segurança o encerramento do laço encerrando-o imediatamente.

A variável `soma` acumula os valores que o usuário digitou e armazena em `num`. O operador `+= soma` esses valores e atribui o resultado a variável `soma`.

Por fim, `printf` exibe a saída do resultado ao usuário. O especificador `%d`, ainda dentro dos parênteses, pede ao compilador que mostre o resultado como número inteiro decimal. Sem ele, o `printf` não saberia interpretar corretamente o conteúdo de `soma`. Depois vem indicada a variável `soma` porque é o valor dela que deve ser exibido.

Return 0, retorno da função `main`, finaliza, indicando o fim da execução da função. A chave final `}` fecha e delimita o corpo da função iniciada com `main`.

Espero ter compreendido e me expressado corretamente. Fiz anotações particulares das aulas não tendo bibliografia adicionada ao conteúdo.

The screenshot shows the OnlineGDB interface. On the left, a sidebar menu includes options like 'Create New Project', 'My Projects', 'Classroom', 'Learn Programming', 'Programming Questions', 'Upgrade', and 'Logout'. The main area displays a C program in a code editor:

```
main.c
21 ****
8
9 #include <stdio.h>
10 int main() {
11     int num;
12     int soma = 0;
13
14     while (num != 0) {
15         printf("Digite um número ou 0 para encerrar: \n");
16         scanf("%d", &num);
17         if (num == 0) {
18             break;
19         }
20         soma += num;
21     }
22     printf("A soma total dos números é: %d\n", soma);
23
24     return 0;
25 }
```

Below the code editor, there's an 'input' field containing the user's input: '67', '3', and '0'. The output window shows the program's response: 'Digite um número ou 0 para encerrar:' followed by the user's inputs, and then 'A soma total dos números é: 70'. The interface also features a top navigation bar with 'Run', 'Debug', 'Stop', 'Share', 'Save', 'Beautify', and download/share icons, and a language selection dropdown set to 'C'. A sidebar on the right contains ads for AWS and Microsoft Azure.

Print da imagem com o resultado do código rodando.