

Exemplo de atividade inicial que um aluno com dificuldades em programação pode realizar sobre o Programa2.c, para iniciar sua familiaridade com a linguagem e com os programas.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <math.h>
```

Informando ao compilador sobre arquivos com definições complementares que este programa pode precisar. Esses arquivos são, em geral, pré-definidos na linguagem.

```
int main(void)
{
    unsigned char cNum;
    int iNum;
```

Declaração das variáveis específicas deste programa. Cada variável tem seu próprio tipo de dados e também tem um nome, que é escolhido pelo programador.

```
cNum = 0;
```

```
iNum = 0;
```

Colocando um valor inicial nas variáveis. Como as variáveis acabaram de ser criadas no programa, elas não possuem um conteúdo específico, que está sendo trocado por zero aqui.

```
while( iNum < 260 )
```

Começo de um looping. Será encerrado qdo iNum receber um valor maior ou igual a 260.

```
{
    printf( "%d %d\n", iNum, cNum );
```

Imprimindo na tela os valores das duas variáveis.

```
cNum = cNum + 1;
```

```
iNum = iNum + 1;
```

Somando um em cada uma das duas variáveis.

```
}
```

```
return 0;
```

```
}
```

Chegando aqui, retorna para a linha do while e faz a comparação do valor de iNum: se for menor que 260, passa mais uma vez pelas 3 linhas entre as duas chaves; se iNum for maior ou igual a 260, sai do looping e chega no return.

Encerra o programa e retorna um código de final OK para o sistema operacional.

O aluno deve comentar (a lápis, em texto livre) TODAS as linhas de código do programa em uma listagem do mesmo que esteja impressa em papel (é importante que esteja impressa em papel). Se tiver alguma dúvida sobre o significado de alguma instrução do programa, o aluno deve primeiro tentar resolver sua dúvida sozinho, pesquisando em livros, apostilas, notas de aula e internet. Só depois de ter feito essa pesquisa individualmente é que ele deve perguntar para alguém (monitor, colega ou professor) a resposta à sua dúvida, caso ainda não a tenha solucionado. Sem isso o aluno nunca vai ganhar autonomia e capacidade de auto-aprendizado. O monitor pode, caso o aluno peça, dar uma olhada nos comentários anotados pelo aluno e fazer comentários, correções e sugestões.

Após comentar e entender o código fonte, o aluno deve realizar um teste de mesa com ele, simulando em papel a execução do programa, anotando o resultado final produzido por aquela simulação. Caso tenha dúvidas ao tentar reproduzir algum trecho do programa, o aluno primeiro deve tentar solucionar essa dúvida por conta própria, pesquisando nas fontes escritas a que tiver acesso. Apenas após ter tentado solucionar por si mesmo sua dúvida, o aluno deve pedir orientação para outras pessoas.

Concluída a simulação via teste de mesa, o aluno deve codificar o programa, compila-lo e executá-lo no ambiente de desenvolvimento. Neste momento podem ocorrer diversas dúvidas de caráter operacional e é importante orientar constantemente o aluno em todas as suas dúvidas sobre o ambiente. O estudante deve executar o programa com os mesmos valores de entrada que utilizou em seus testes de mesa e comparar os resultados do programa com aqueles obtidos na simulação inicial. Tentar identificar os erros cometidos e entender a sua origem.

Após essas atividades, o aluno deve tentar criar pequenas modificações no código fonte, visando aumentar seu domínio sobre a linguagem.