

Senhores, como puderam perceber, o Susy é bem chato com respeito a saída do programa. Isso acontece porque ele compara caracter a caracter a sua saída com a saída esperada. Infelizmente, o lab00 também não ajudou muito com uma saída cheia de detalhes.

Para que vocês não tenham mais este tipo de problema nos labs futuros, sugerimos que façam o seguinte:

1. Leiam o lab com atenção.
2. Entendam o que está sendo dado como entrada e o que se espera de saída (veja os exemplos do enunciado!)
3. ANTES de resolver o problema propriamente, escrevam já no .c o scanf para ler os valores de entrada e o printf para imprimir os valores de saída (substituem o valor do %d por 0, por exemplo).

Neste momento, o lab de vocês seria algo do tipo:

```
#include <stdio.h>

int main() {
    int valor;

    printf("Valor a ser retirado: ");
    scanf("%d", &valor);

    // FIXME: Calcular o valor aqui

    // FIXME: substituir pelos valores corretos
    printf("Seria(m) emitida(s) %d nota(s): %d nota(s) de 100.00, "
           "%d nota(s) de 50.00, %d nota(s) de 10.00, %d nota(s) de 5.00 "
           "e %d nota(s) de 1.00.", 0, 0, 0, 0, 0, 0);

    return 0;
}
```

Aproveitem para compilar e CHECAR que vocês conseguem ler corretamente e que a impressão está exatamente como esperado. Em caso de erro na compilação, vejam a saída do compilador: geralmente ele te diz onde está e qual é o erro.

Agora vem a parte em que vocês esquecem o programa e PENSAM na solução (pode ser com a ajuda de um papel). No caso do lab, as perguntas chaves que vocês tinham que fazer eram:

- * Como faço para calcular quantas notas de 100, 50, 10, 5 e 1 precisarei?
- * E como faço esse cálculo de forma a retornar o menor número de notas?
- * Como faria isso com papel e caneta?
- * E como ensinaria minha avó a fazer isso?

Feitas essas perguntas e, possivelmente, após alguns testes, vocês se dão conta de que a resposta é: utilize o máximo que puder de notas de 100, então o máximo de 50 e assim por diante até as notas de 1.

PRONTO! O lab está feito:

```
int main() {
    int valor, notas100, notas50, notas10, notas5, notas1, total_notas, resto;

    printf("Valor a ser retirado: ");
    scanf("%d", &valor);
```

```

notas100 = valor / 100;
resto = valor % 100; // ou resto = valor - notas100 * 100;

notas50 = resto / 50;
resto = resto % 50; // ou resto %= 50; ou resto -= notas50 * 50;

notas10 = resto / 10;
resto = resto % 10; // ou os 'ous' acima

notas5 = resto / 5;
resto %= 5;

notas1 = resto;

total_notas = notas100 + notas50 + notas10 + notas5 + notas1;

printf("Seria(m) emitida(s) %d nota(s): %d nota(s) de 100.00, "
      "%d nota(s) de 50.00, %d nota(s) de 10.00, %d nota(s) de 5.00
"
      "e %d nota(s) de 1.00.", total_notas, notas100, notas50,
notas10,
      notas5, notas1);

return 0;
}

```

Notem, no exemplo acima, como linhas em branco e espaços foram usados para facilitar a leitura do código. Em várias das soluções, vimos coisas do tipo:

```
x=y/50;
```

Comparem agora com:

```
notas50 = resto / 50;
```

Vejam como os espaços e o uso de nomes significativos para as variáveis (notas100, notas50, ...) facilitam e MUITO o entendimento do que se está querendo fazer.

Algumas pessoas perderam um tempo tentando descobrir porque o programa não funcionava porque usaram variáveis a, b, c, ..., trocaram uma delas em uma das contas (era para ser um 'b' e não um 'c') e não conseguiam ver o erro.

Outras pessoas usaram 'if' para testar se o valor da divisão era < 1. Essas pessoas teriam o lab ZERADO se ele valesse nota, pois o enunciado dizia explicitamente:

"Não serão aceitas soluções contendo estruturas não vistas em sala (para este laboratório, poderão ser utilizadas apenas operações aritméticas e a única biblioteca permitida para inclusão é a stdio.h)."

Além do mais, o if nem era necessário, pois a divisão entre números inteiros SEMPRE retorna um número inteiro. Então, qualquer valor inteiro menor que x, quando dividido por um inteiro x, dará 0 (zero).

Mais três lembretes:

1. Indentem o código de vocês e não percam nota por bobeira! Exemplos como a seguir perderão até 2 pontos:

```
#include<stdio.h>
int main(){
```

```
int valor, a, b, c;  
scanf("%d",&valor);  
a=b/100;  
c=b%100;  
}
```

2. Não se esqueçam do cabeçalho do programa!! O arquivo .c de vocês deve começar assim:

```
/*  
    Nome: Fulano da Silva  
    RA: XXXXXX  
  
    Descrição dos objetivos do programa.  
*/  
  
#include <stdio.h>  
  
[...]
```

3. Após o período de submissão, no link "Soluções", vocês poderão ver um exemplo de solução esperada. Recomendo que vejam o arquivo e comparem a solução com a de vocês --- pode ser que a solução fosse mais fácil que você imaginava!

Boa sorte nos próximos labs!