

Universidade de Brasília
Departamento de Ciência da Computação

116432 - Software Básico - Turma B

1. (1.5 pontos) O Código abaixo tem exatamente 3 erros. Indique quais são (sublinhando ou ressaltando de alguma forma a linha no código) e coloque o tipo de erro (léxico, sintático ou semântico). Pode também copiar SOMENTE as linhas com erro na folha de resposta e indicar o tipo nessa resposta.

```
#include <stdio.h>
int main()
{
    int array[100], n, d, swap;
    int C&;

    printf("Enter number of elements\n");
    scanf("%d", &n);

    printf("Enter %d integers\n", n);

    for (c = 0; c < n; c++)
        scanf("%d", &array[c]);

    for (c = 0 ; c < n - 1; c++)
    {
        for (d = 0 ; d < n - c - 1; d++)
        {
            if (array[d] > array[d+1]) /* For decreasing order use '<' instead of '>' */
            {
                swap    = array[d];
                array[d] = array[d+1];
                array[d+1] = swap;
            }
        }
    }
    printf("Sorted list in ascending order:\n");

    for (c = 0; c < n)
        printf("%d\n", array[c]);
    return 0;
}
```

Universidade de Brasília
Departamento de Ciência da Computação

2. Dado o programa abaixo.
- (2 pontos) Mostre como ficaria a MNT e a MDT.
 - (2 pontos) Mostre a tabela de símbolos assumindo que foi utilizado p algoritmo de passagem única, indicando as listas de pendencia (note que é necessario primero resolver as macros).

```
SECTION TEXT
M1: MACRO &A,&B,&C
COPY &A,&B
INPUT &C
OUTPUT &B
ENDMACRO
M2: MACRO &A,&B,&C
OUTPUT &A
COPY &B,&C
COPY &A,&B
ENDMACRO
COPY ZERO,OLDER
        M1 ONE,OLD,LIMIT
FRONT:  LOAD OLDER
        ADD OLD
        STORE NEW
        SUB LIMIT
        JMPP FINAL
        M2 NEW,OLD,OLDER
        JMP FRONT
FINAL: OUTPUT LIMIT
        STOP
SECTION DATA
ZERO: CONST 0
ONE: CONST 1
OLDER: SPACE
OLD: SPACE
LIMIT: SPACE
```

Universidade de Brasília
Departamento de Ciência da Computação

3. Faça um programa utilizando o Assembly hipotético visto em sala de aula que receba do usuário três números. E faça as seguintes operações: (i) Salve os três números em três labels diferentes; (ii) compare os números, de forma de identificar o maior e menor deles; (iii) mostre na tela os 3 números ordenados, mostrando primeiro a menor número depois o intermediário e finalmente o maior número e (iv) perguntar para o usuário se deseja repetir todo o processo, caso negativo terminar o programa.
- a. (1.5 pontos) Elabore o programa em 2 (dois) módulos. O primeiro módulo deve fazer a tarefa (i), pular para o segundo módulo onde deve ser feito as tarefas (ii) e (iii), pular de volta para o módulo 1 e fazer a tarefa (iv). A seção de dados deve sempre ir no final
 - b. (1.5 pontos) Mostre as tabelas de uso, símbolo e definições de cada um dos módulos
 - c. (1.5 ponto) Ligue o programa mostrando o código máquina (do assembly inventado), indicando o fator de correção e a tabela global de definições. O código máquina deve ser mostrado seguindo o formato visto em sala de aula.

Universidade de Brasília

Departamento de Ciência da Computação

Tabela 1: Tabela de Código Montador

Instruções			
Mnemônico	Operandos	Código	Tamanho
ADD	1	01	2
SUB	1	02	2
MULT	1	03	2
DIV	1	04	2
JMP	1	05	2
JMPN	1	06	2
JMPP	1	07	2
JMPZ	1	08	2
COPY	2	09	3
LOAD	1	10	2
STORE	1	11	2
INPUT	1	12	2
OUTPUT	1	13	2
STOP	0	14	1
Diretivas			
SPACE	0 ou 1	-	variável
CONST	1	-	1
EXTERN	0	-	0
PUBLIC	0	-	0
BEGIN	0	-	0
END	0	-	0
MACRO	variável	-	0
ENDMACRO	0	-	0