

## Lista de Exercícios de Organização de Computadores

Prof. Ronaldo Luiz Alonso

1) Traduza os códigos seguintes para linguagem assembly x86 64 bits:

a) empregado2.c

```
struct empregado {  
    float salario;  
    char  nome[64];  
    int   idade;  
    char  sexo;  
};  
  
struct empregado emp = {  
    1000000.0,  
    "Dr Milionário",  
    100,  
    'M'};
```

b) empregado2.c (continuação)

```
int main (void)  
{  
    printf("%f\n",emp.salario);  
    printf("%s\n",emp.nome);  
    printf("%d\n",emp.idade);  
    printf("%c\n",emp.sexo);  
  
    printf("Entre com o salario:");  
    scanf("%f",&(emp.salario));  
    printf("Entre com o nome:");  
    scanf("%s",&(emp.nome));  
    printf("Entre com a idade:");  
    scanf("%d",&(emp.idade));  
  
    printf("Novos valores:\n");  
  
    printf("%f\n",emp.salario);  
    printf("%s\n",emp.nome);  
    printf("%d\n",emp.idade);  
    printf("%c\n",emp.sexo);  
  
    return 0;  
}
```

c) bubble.c (parte 1)

```
int vetor[10] = {10,9,8,7,6,5,4,3,2,1};  
int tamanho = 10;  
int aux;
```

d) bubble.c (parte 2)

```
int main(void){  
  
    for (i = 0; i < MAX; i++)  
    {  
        printf("Entre com um numero:");  
        scanf ("%d",&vetor[i]);  
    }  
  
    for(int i=tamanho-1; i >= 1; i--) {  
  
        for( int j=0; j < i ; j++) {  
  
            if(vetor[j]>vetor[j+1]) {  
  
                aux = vetor[j];  
                vetor[j] = vetor[j+1];  
                vetor[j+1] = aux;  
  
            }  
        }  
    }  
}
```

e) bubble.c (parte 3)

```
for( int r = 0; r < 10; ++r){  
    printf("%d\n",vetor[r]);  
}  
return 0; |  
}
```

f) empregado.c

```
#include <stdio.h>

struct empregado {
    double salario;
    int idade;
};

struct empregado emp;

int main (void)
{
    printf("Entre com o salario:");
    scanf("%lf",&emp.salario);
    printf("Entre com a idade:");
    scanf("%d",&emp.idade);
    printf("salario: %lf\n",emp.salario);
    printf("idade: %d\n",emp.idade);
    return 0;
}
```

g) float1.c

```
#include <stdio.h>
double a,b;
double sum, dif, mult, div;

int main()
{
    printf("Entre com o valor de a:");
    scanf ("%f",&a);
    printf("Entre com o valor de b:");
    scanf ("%f", &b);
    sum = a+b;
    dif = a - b;
    mult = a * b;
    div = a/b ;
    printf("Soma: %f\n",sum);
    printf("Diferença: %f\n",dif);
    printf("Multiplicação: %f\n",mult);
    printf("Divisao: %f\n",div);
}
```

h) prime.c

```
int main()
{
    int n, i = 3, count, c;

    printf("Enter the number of prime numbers required\n");
    scanf("%d",&n);

    if ( n >= 1 )
    {
        printf("First %d prime numbers are :\n",n);
        printf("2\n");
    }

    for ( count = 2 ; count <= n ; )
    {
        for ( c = 2 ; c <= i - 1 ; c++ )
        {
            if ( i%c == 0 )
                break;
        }
        if ( c == i )
        {
            printf("%d\n",i);
            count++;
        }
        i++;
    }
    return 0;
}
```

2) Escreva uma função em assembly chamada “maximo”, isto é, crie um arquivo maximo.asm que contenha essa função e que tal função possa ser chamada a partir do seguinte programa em C:

max.c

```
#include "stdio.h"

long int maximo(int a , int b, int c);

long int main ()
{
    long int x,y,z;
    long int m;
    printf("Entre tres valores inteiros: ");
    scanf ("%ld %ld %ld",&x,&y,&z);
    m = maximo(x,y,z);
    printf("O maximo eh: %ld\n",m);
    return 0;
}
```