

Universidade Federal de Minas Gerais
Colégio Técnico - COLTEC



Prof. Márcio Fantini

Avaliação 2 - Módulo 2 (15pts)

Grupo:

Data: 05/10/2021

Cauã Magalhães Pereira;

Fernando Alves de Freitas;

Pedro Guilherme Andrade Salgado;

Gabriel Lyan Barbosa de Assis.

1)

a) 2048 (Decimal) = 800 (Hex) endereços

b)

Menor: 0000 (Decimal) 000 (Hex) 000000000000 (Binário)

Maior: 2047 (Decimal) 7FF (Hex) 1111111111 (Binário)

c) FF (Hex), 255 (Decimal), 1111 1111 (Binário)

d) A2 (Hex) = 0x0A2

e) São necessários pelo menos 2 dígitos

f)

Endereço (Hex)	Conteúdo
0x0A2	T
0x0A3	E
0x0A4	S
0x0A5	T
0x0A6	E
0x0A7	V
0x0A8	E
0x0A9	T
0x0AA	O
0x0AB	R
0x0AC	\0

2)

- a) $*p * *q \rightarrow 2 * 272 \rightarrow 544(\text{dec})$
- b) $\&c \rightarrow X+4$
- c) $*p - *(&a) \rightarrow 2 - 2 \rightarrow 0$
- d) $*(p+2) \rightarrow c \rightarrow \text{void (lixo)}$
- e) $*q \rightarrow b \rightarrow 272(\text{dec})$ ou $0x0110$
- f) $*a \rightarrow$ O programa não irá rodar, pois a não é um pointer. Ao usar o $(*a)$ o programa não iria rodar, por "a" não ser um pointer

Endereço (Hex)	Conteúdo (Dec)	variáveis
X	0	a
X+1	2	
X+2	0	b
X+3	6	
X+4		c
X+5		
X+6	0	d
X+7	0	
X+8	0	
X+9	'a'	e

Tipo inteiro ocupa 2 bytes, do tipo char ocupa 1 byte e tipo float ocupa 3 bytes.

```
int a,b,c,*p,*q,*r;
char e ='a';
float d=0;
a=2;b=0x0110;
p=&a;q=&b;
```

3)

- a) $\&a[3] \rightarrow X+6$
- b) $*(&a[0]) \rightarrow 0$
- c) $a[3] + *(p+2) \rightarrow 257$
- d) $\&a \rightarrow X$
- e) $p \rightarrow X$
- f) $*(q+1) \rightarrow 'b'$

4)

a)

Código do Loop da Questão 4.a)

```
int i;

for(i=0; i<40; i= i + 2){
    printf("%d\n", i);
}
```

b)

Código do Loop da Questão 4.b)

```
float soma = cos(M_PI);
int i;

for(i=2; i<21; i++){

    if(i % 2 == 0){
        soma = (float)soma + i*sin(M_PI/i);
    }

    if(i % 2 == 1){
        soma = (float)soma + i*cos(M_PI/i);
    }

}
```

c)

Código do Loop da Questão 4.c)

```
int numero=-1, quadrado;

while(numero < 0){

    printf("Digite um numero: \n");
    scanf("%d", &numero);

    if(numero < 0){

        quadrado = pow(numero,2);

        printf("O valor de x eh: %d\nE o valor de x ao quadrado eh: %d\n\n", numero, quadrado);

    }else{
        printf("O numero digitado nao eh negativo! ");
    }

}
```

- 5) A saída da função main() é 0, já a da função1() é o menor valor de arr[] na posição 0 e o maior na posição 1.

Código corrigido:

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

int * funcao1(int arr[],int n){

    int i,max,min;
    // 1) defina corretamente v
    int *v;

    // 2) aloque corretamente
    // memoria para v
    v = (int*)malloc(2*sizeof(int));

    max = arr[0]; min = max;
    for (i = 0; i < n; i++) {
        if (arr[i] > max)
            max=arr[i];
        if (arr[i] < min)
            min=arr[i];
    }
    v[0] = max;*(v+1) = min;
    return v; }

int main() {

    int arr[] = { 52, 3, 4, 10, 40 };
    // 3) declarar corretamente Mm;
    int *Mm;

    //4) alocar memoria adequada para Mm-
    //(dica: repare como Mn deve ser usado)

    //Não aloquei "Mm", pois a funcao retorna um endereço para o mesmo,
    //alocar outro valor para ele no main representaria um desperdicio
    int n = sizeof(arr) / sizeof(arr[0]);

    //5) chame a funcao, passando os
    // parametros corretos
    // e recebendo o retorno em Mm

    Mm = funcao1(arr,n);

    //6) mostre os valores armazenados em Mm

    printf("min = %i e max = %i\n",Mm[1],Mm[0]);

    free(Mm);

    return 0;
```

}
