**REVISÃO ESTRUTURA DE DADOS II**

**Atividade 01**

Do mesmo modo que dizemos que o endereço da variável x é E2, podemos dizer que o endereço do array a é E100. Isto é, o endereço do array é o endereço do primeiro byte que o array ocupa.

De fato, quando declararmos: int a[10]; **int a[10]; int \*p; p = a; \*p = 10;** é equivalente:

1. a[0] = 10 ;
2. \*a[0] = 10;
3. a= 10;
4. &a = 10;
5. NDA

**Atividade 02**

Marque a alternativa Correta:

Se você quiser usar o conteúdo do ponteiro \*p 15 posições adiante, deverá escrever:

1. \*(p+15);
2. \*P[15];
3. &p[15];
4. (\*p+15);
5. NDA;

**Atividade 03**

Considerando o programa abaixo, responda:

**#include <stdio.h>**

**#include <stdlib.h>**

**main (void)**

**{**

**int \*p, a, \*i;**

**i=&a;**

**p=&a;**

**a = 30;**

**p=(int \*)malloc(a\*sizeof(int));**

**if (!p)**

**{**

**printf ("\*\* Erro: Memoria Insuficiente \*\*");**

**exit (1);**

**}**

**}**

Quantos bytes ocupam o ponteiro p?

Resposta: \_\_\_\_\_\_\_

Qual é resultado de impressão de \*(i) no final do programa?

Resposta: \_\_\_\_\_\_\_

**Atividade 04**

Considerando o código existente do programa abaixo, Quantos acessos ao vetor são necessários para localizar o valor 20?

**#include <stdio.h>**

**#include <stdlib.h>**

**#define TAMANHO 20**

**main()**

**{**

**int x, i=0, vetor[20] = {3,5,6,7,8,9,10,12,14,16,19,20,21,22,33,35,37,39,44,50};**

**printf("Qual Numero deseja localizar?:");**

**scanf("%d",&x);**

**int meio=0, esquerda = -1, direita = TAMANHO;**

**while (esquerda < direita-1)**

**{**

**int meio = (esquerda + direita)/2;**

**if (vetor[meio] < x)**

**esquerda = meio;**

**else**

**direita = meio;**

**}**

**if (vetor[direita]==x)**

**printf("Valor Localizado");**

**else**

**printf("Valor NAO Localizado");**

**}**

**Atividade 05**

O que imprime o programa a seguir?

**void func(int aa, int bb)**

**{**

**int temp;**

**temp = aa;**

**aa = bb;**

**bb = temp;**

**}**

**int main()**

**{**

**int a = 10, b = 20;**

**func(a, b);**

**printf("a = %d, b = %d", a, b);**

**}**

**Atividade 06**

O que imprime o programa a seguir?

**void func(int \*aa, int \*bb)**

**{**

**int temp;**

**temp = \*aa;**

**\*aa = \*bb;**

**\*bb = temp;**

**}**

**int main()**

**{**

**int a = 10, b = 20;**

**func(&a, &b);**

**printf("a = %d, b = %d", a, b);**

**}**

**Atividade 07**

Considerando o programa abaixo, qual será o valor de I no final do programa?

**#include <stdlib.h>**

**#include <stdio.h>**

**void somaUm(int &x)**

**{**

**x++;**

**}**

**main()**

**{**

**int i=0;**

**while (i<10)**

**{**

**somaUm(i);**

**}**

**printf("valor de i : %d \n",i);**

**}**

**Atividade 08**

Considerando o programa abaixo, qual será o valor de I no final do programa?

**#include <stdlib.h>**

**#include <stdio.h>**

**void somaUm(int x)**

**{**

**x++;**

**}**

**main()**

**{**

**int i=0;**

**while (i<10)**

**{**

**somaUm(i);**

**}**

**printf("valor de i : %d \n",i);**

**}**

**Atividade 09**

Considerando o programa abaixo, qual será o valor de I no final do programa?

**#include <stdlib.h>**

**#include <stdio.h>**

**void somaUm(int \*x)**

**{**

**(\*x)++;**

**}**

**main()**

**{**

**int i=0;**

**while (i<10)**

**{**

**somaUm(&i);**

**}**

**printf("valor de i : %d \n",i);**

**}**

**Atividade 10**

Considerando a estrutura Pessoa, como é possível acessar via ponteiro a propriedade nome?

**typedef struct Cpessoa**

**{**

**char nome[20];**

**int idade;**

**} ;**

**int main(void)**

**{**

**Cpessoa aluno;**

**Cpessoa \*paluno;**

**paluno = &aluno;**

**...**

**}**

**Atividade 11**

Considerando a estrutura Pessoa, como é possível acessar a propriedade nome, sem usar ponteiro?

**typedef struct Cpessoa**

**{**

**char nome[20];**

**int idade;**

**} ;**

**int main(void)**

**{**

**Cpessoa aluno;**

**Cpessoa \*paluno;**

**paluno = &aluno;**

**...**

**}**

**Atividade 12**

Considerando o algoritmo abaixo e analisando o seu comportamento, qual será o conteúdo do arquivo **Arquivo.txt** no final da execução?

**main(){**

**int i;**

**unsigned int ex;**

**FILE \*txt;**

**ex = ((unsigned) time(NULL)) ;**

**srand(ex) ;**

**i = 0;**

**while(i < 100)**

**{**

**txt=fopen("Arquivo.txt", "w");**

**float c = rand() % 100;**

**fprintf(txt, "%d - Execução - %d\n",i, i+1);**

**++i;**

**fclose(txt);**

**}**

**}**