



ATIVIDADE

1. Como ficam os valores no final da execução do trecho de programa?

a)

i	j	p	q	p	i
15	?	?	?		15
1080	1082	1084	1086		

```
int i=15,j,*p,*q;  
p = &i;  
*p=20;  
j=2 x *p;  
q=&i;  
*p=*q-1;  
q=&j;  
*p=*q-1;
```

b)

```
void main() {  
    int x, y, *p; y = 0;  
    p = &y;  
    x = *p;  
    x = 4;  
    (*p)++;  
    --x;  
    (*p) += x;  
}
```

x=? y=? *p=?

2. Encontre os erros:

a)

```
void main() {  
    int x, *p;  
    x = 100;  
    p = x;  
    printf("Valor de p: %d.n", *p);  
}
```

b)

```
void troca (int *i, int *j) {  
    int *temp;  
    *temp = *i;  
    *i = *j;  
    *j = *temp;  
}
```

c)

```
main(){  
  
    char *a, *b;  
    a = "uva";  
    b = "carro";  
  
    if (&a==&b)  
        printf("são iguais");  
    else{  
        if (a < b)  
            printf ("%s vem antes de %s no dicionário", a, b);  
        else  
            printf ("%s vem depois de %s no dicionário", a, b);  
    }  
}
```

3. Sobre o programa a seguir:

```
#include <stdio.h>  
#include <iostream>  
using namespace std;  
  
typedef struct no{  
    int item;  
    struct no *prox;  
}No;  
  
int main(){  
    No v[4]={ {3,v+2},{4,NULL},{10,v+1}};  
    for (No *p=v;p!=NULL;p=p->prox){  
        cout<<p->item<<endl;  
    }  
}
```

a. Que valores serão impressos ao final do programa?

b. Considere os seguintes nós n1 e n2:

```
No *n1=(No*)malloc(sizeof(No));  
n1->item = 1;  
n1->prox = NULL;
```

```
No *n2=(No*)malloc(sizeof(No));
n2->item = 4;
n2->prox = NULL;
```

b.1. Como encadear os nós de forma a criar uma lista na sequência {n1,n2,NULL}

b.2. Como percorrer os nós e imprimir todos os itens da lista?

OBS: saída deve ser : 1 4

c. Considere os nós:

```
No *n1=new No();
n1->item = 10;
n1->prox = NULL;

No *n2=new No();
n2->item = 3;
n2->prox = NULL;

No *n3=new No();
n2->item = 4;
n2->prox = NULL;
```

c.1. Como encadear os nós de forma a criar uma lista ordenada pelo item (3,4,10).

c.2. Percorra os nós e imprima todos os itens da lista.

OBS: SAÍDA DEVE SER: 3 4 10

4. Para o código abaixo:

4.1. Crie um programa que defina os seguintes dados do tipo Pessoa:

cpf	nome	prox
1111	Fabio	NULL
2222	katia	NULL
4444	Jonatas	NULL

4.2. Como encadear os dados de maneira a ter uma lista na seguinte ordem de cpf:

{444,111,222}

4.3. Percorra os dados e imprima nome e cpf de cada um.

```
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include <iostream>
using namespace std;

typedef struct pessoa{
    int cpf;
    string nome;
    pessoa *prox;
}Pessoa;
```

```
int main(){  
    }  
}
```

5. Complete o código abaixo:

```
#include <stdio.h>  
#include <iostream>  
using namespace std;  
  
typedef struct aluno{  
    int mat;  
    string nome;  
    aluno *prox;  
}Aluno;  
  
Aluno *no(int mat,string nome){  
    Aluno *novo = new Aluno();  
    novo->mat = mat;  
    novo->nome = nome;  
    novo->prox = NULL;  
    return novo;  
}  
Aluno *inicio = NULL;  
Aluno *fim = NULL;  
  
void incluirNoFinal(Aluno *n){  
  
}  
  
//inclui o elemento no inicio da lista  
void incluirNoInicio(Aluno *n){  
  
}  
  
//retorna o elemento se encontrado  
//se nao encontrado retorna NULL  
Aluno *procura(int mat){  
  
}  
  
//mostra TODOS os elementos da lista  
void mostra(){  
  
}  
}  
main(){  
    Aluno *novo = no(1,"maria");  
    incluirNoFinal(novo);  
}
```

```
    novo = no(2,"katia");  
    incluirNoFinal(novo);  
    mostra();  
}
```