**Curso Tecnólogo em Sistemas para Internet**



**Disciplina de Programação Estruturada**

**Avaliação**

Nome: Everton Oliveira Fernandes

Turma: TSI-2AN Data: 01/06/2016

1. Sobre ponteiros, responda:
   1. O que são? Cite um exemplo de sua utilidade.
   2. Explique os operadores de ponteiros & e \*.
   3. Mostre como atribuir valores para uma variável e para um vetor através de um ponteiro.
   4. Considerando p um ponteiro, explique a diferença entre entre os comandos (\*p)++ e p++.
   5. Qual a diferença entre ponteiros do tipo char e int?
2. Sobre funções, responda:
   1. O que são? Cite pelo menos três benefícios de sua utilização.
   2. O que são passagem de parâmetros para funções? Explique como funciona os dois tipos. Mostre um exemplo de cada.
   3. Para que serve o comando return? Explique como funciona o retorno das funções.
   4. Mostre um exemplo de uma função void, int e char, bem como suas respectivas chamadas.
3. Analise o trecho de código abaixo:

int aleatorio(int inf, int sup){ srand(time(NULL));

return rand()%(sup-inf+1)+inf;

}

main(){

int a,b;

scanf("%i %i",&a,&b);

printf("%i",aleatorio(a,b));

}

Reescreva o trecho de código, transformando a função *aleatório* em um procedimento (função sem retorno), e removendo a passagem de parâmetros, fazendo com que a comunicação entre as funções *main* e *aleatorio* se dê exclusivamente pelo uso de variáveis globais.

1. Analise o trecho de código abaixo, explicando o que acontece em cada linha, e diga o que aparece na tela ao fim da execução:

int f[5]={1,2,3,4,5},\*p,\*q,i,j; p=f; q=f+4;

for (i=0 ; i<5 ; i++)

\*(p+i)=\*(q-i);

\*p = \*(q-2);

\*(q-\*p)-=\*p+\*q;

q--; p++;

\*p+=\*q;

f[3] = 2+\*q\*(\*p);

for (p=f ; p<=q; p++) printf("%i",\*p);

1. Sobre alocação dinâmica, responda:
   1. O que é, e para que serve?
   2. Crie um trecho de código que aloque um espaço para 5 números inteiros, e preencha-os com valores informadores pelo usuário. Desaloque o espaço alocado ao fim.
2. Observe o trecho de código abaixo:

int remove(int vet[], int t){ t--;

int r = vet[t];

vet = (int\*)realloc(vet,sizeof(int)\*t); return r;

}

Supondo que a função *remove* deve receber um conjunto de números *vet* de tamanho *t*, retornar o último elemento deste conjunto e reduzir em 1 o espaço alocado para estes números (afinal, a função remove um elemento), aponte os erros da função, corrigindo-os, bem como apresente como seria a chamada desta função. A função não apresenta erros de lógica, somente no uso de ponteiros.

1. Observe o código abaixo:

int maimen(int v[t]){ int i,m,n;

for (i=0 ; i<t ; i++){

if (v[i] > m || i==0) m = v[i];

if (v[i] < m || i==0) n = v[i];

}

return m,n;

}

Considerando que a função deva receber um conjunto numérico *v* de tamanho *t*, e verificar e atribuir por referência o maior e menor elemento deste conjunto, aponte os erros no código e corrija o mesmo.