

Nome: Gabriel Kenji Inoue

PONTEIROS

1. Quais serão os valores de x, y e p ao final do trecho de código abaixo?

```
int x, y, *p;  
y = 0;  
p = &y; /*p = 0  
x = *p; //x = 0  
x = 4; // x = 4  
(*p)++; /*p = 1, y = 1  
--x; // x = 3  
(*p) += x; /*p = 4, y = 4
```

R: X = 3, Y = 4, *P = 4

2. Os programas (trechos de código) abaixo possuem erros. Qual(is)? Como deveriam ser?

a)

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

R: O erro nesse caso se deve ao fato de P ser um ponteiro e ele recebe o valor de x, não o endereço. O correto seria: p = &x;

b)

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

R: O erro se deve ao fato da não necessidade da variável temp ser um ponteiro, o correto seria ele ser um int comum. O correto seria: int temp;

c)

```
char *a, *b;  
a = "abacate";  
b = "uva";  
if (a < b)  
    printf ("o/os vem antes de o/os no dicionario", a, b);  
else  
    printf ("o/os vem depois de o/os no dicionario", a, b);
```

R: Esse código contém vários erros na realidade. Primeiramente a variável “a” e “b” são declaradas como ponteiros e ao realizarmos a comparação, estamos comparando nesse caso o endereço das variáveis em si, não o conteúdo. Para compararmos corretamente precisamos incluir a biblioteca string.h e realizar a função strcmp, que mostrará a ordem alfabeticamente, ficando:

```
int compara = strcmp (a,b)  
if(compara < 0) {...}, else if(compara > 0) {...}, else {...}
```

3) Suponha que os elementos do vetor `v` são do tipo `int` e cada `int` ocupa 8 bytes no seu computador. Se o endereço de `v[0]` é 55000, qual o valor da expressão `v + 3`?

R: Como cada posição de endereço do vetor de `int` “avança” a cada 8 bytes, o `v[3]` tem o valor 55024, pois seria $55000 + 24$ (já que $8 * 3 = 24$).

4) Escreva uma função `mm` que receba um vetor inteiro `v[0..n-1]` e os endereços de duas variáveis inteiras, digamos `min` e `max`, e deposite nessas variáveis o valor de um elemento mínimo e o valor de um elemento máximo do vetor. Escreva também uma função `main` que use a função `mm`.

R: `#include <stdio.h>`

```
void mm(int n, int *min, int *max) {
    int vet[n];
    int i;

    for(i = 0; i < n; i++) {
        printf("Digite o valor da posicao %d: ", i);
        scanf("%d", &vet[i]);
        if(i == 0) {
            *max = vet[i];
            *min = vet[i];
        }
        else if (*max < vet[i]) {
            *max = vet[i];
        }
        else if (*min > vet[i]) {
            *min = vet[i];
        }
    }
}

int main() {
    int n;
    int max, min;
    printf("Digite o tamanho do seu vetor: ");
    scanf("%d", &n);

    mm(n, &min, &max);

    printf("O menor valor eh: %d\nO maior valor eh: %d", min, max);
    return 0; }
```

5) Suponha que v é um vetor. Descreva a diferença conceitual entre as expressões $v[3]$ e $v + 3$.

R: $v[3]$ significa que o vetor está na 4ª posição ($v[0]$, $v[1]$, $v[2]$, $v[3]$), já a expressão $v + 3$ está relacionada com aritmética de ponteiros, em que avançamos 3 endereços de memórias a partir da posição de v .

6) (sem usar o computador) Qual o conteúdo do vetor a depois dos seguintes comandos.

```
int a[99];
for (i = 0; i < 99; ++i)
    a[i] = 98 - i;
for (i = 0; i < 99; ++i)
    a[i] = a[a[i]];
```

R: $a[0] = 98$, $a[1] = 97$, ..., $a[97] = 1$, $a[98] = 0$

$a[0] = a[98] = 0$, $a[1] = a[97] = 1$, ..., $a[97] = a[1] = 1$, $a[98] = a[0] = 0$.

Resumidamente, a função faz com que a posição $a[0]$ até $a[48]$ tenha os valores de suas posições, contudo, com o novo valor definido, a outra “metade maior” receberá também os valores alterados, ficando com uma simetria final, isto é, a posição 98 receberá 0, a 97 receberá 1, 96 receberá 2, até chegar a posição 49, que receberá 47.

7) Escreva uma função chamada *troca* que troca os valores dos parâmetros recebidos. Sua assinatura deve ser:

```
void troca(float *a, float *b);
```

R:

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

```
void troca(float *a, float *b) {
```

```
    float temp;
```

```
    temp = *a;
```

```
    *a = *b;
```

```
    *b = temp; }
```

```
int main() {
```

```
    float a, b;
```

```
    printf("Digite o primeiro número: ");
```

```
    scanf("%f", &a);
```

```
    printf("Digite o segundo número: ");
```

```
    scanf("%f", &b);
```

```
    troca(&a, &b);
```

```
printf("O primeiro número agora é: %f, e o segundo é %f", a, b);  
  
return 0; }
```

8) Crie uma função que receba uma string como parâmetro (de tamanho desconhecido) e retorne uma cópia da mesma. A assinatura da função deve ser:

```
char *strcpy(char *str);
```

```
R: #include <stdio.h>
```

```
#include <string.h>
```

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

```
char *strcpy(char *str) {  
    int tamanho = strlen(str);  
    int i;  
    char *copia = (char *)malloc((tamanho + 1) * sizeof(char));  
  
    for (i = 0; i <= tamanho; i++) {  
        copia[i] = str[i];  
    }  
    return copia; }
```

```
int main() {  
    char palavra[50];  
    char *copiapalavra;  
    printf("Digite uma palavra: ");  
    scanf("%s", palavra);  
  
    copiapalavra = strcpy(palavra);  
  
    printf("String Original: %s\n", palavra);  
    printf("String copiada: %s\n", copiapalavra);  
    free(copiapalavra);  
    return 0; }
```

9) Escreva uma função que recebe como parâmetros um vetor de inteiros v, o número de elementos dele N e ponteiros para variáveis nas quais devem ser armazenados os valores máximo e mínimo do vetor. Sua assinatura deve ser:

```
void maximoMinimo(int *v, int N, int *maximo, int *minimo);
```

R:

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

```
void maximoMinimo(int *v, int N, int *maximo, int *minimo) {  
    int i;
```

```
    for(i = 0; i < N; i++) {  
        printf("Digite o valor da posicao %d: ", i);  
        scanf("%d", &v[i]);  
        if(i == 0) {  
            *maximo = v[i];  
            *minimo = v[i];  
        }  
        else if (*maximo < v[i]) {  
            *maximo = v[i];  
        }  
        else if (*minimo > v[i]) {  
            *minimo = v[i];  
        }  
    }  
}
```

```
int main() {  
    int n;  
    int maximo, minimo;  
    printf("Digite o tamanho do seu vetor: ");  
    scanf("%d", &n);  
    int vet[n];  
  
    maximoMinimo(vet, n, &maximo, &minimo);  
  
    printf("O menor valor eh: %d\nO maior valor eh: %d", minimo, maximo);  
    return 0; }
```

10) Qual o resultado do código abaixo? Explique cada linha.

```
int x = 100, *p, **pp;  
p = &x; // *p = 100  
pp = &p; // **p = *p = 100  
printf("Valor de pp: %d\n", **pp); // Saída = 100
```

R: O código dará o valor 100, pois como **pp é um ponteiro de um ponteiro, então ele pega o endereço de p, que pega o endereço de x, fazendo com que:

```
**p = *p = x = 100
```

- 11) Escreva uma função que recebe uma string de caracteres e uma letra e devolve um vetor de inteiros contendo as posições (índices no vetor da string) onde a letra foi encontrada e um inteiro contendo o tamanho do vetor criado (total de letras iguais encontradas). Utilize o retorno de um vetor para retornar os índices e um ponteiro para guardar o tamanho do vetor.

R: #include <stdio.h>

#include <stdlib.h>

```
int *acharPosicao(char *palavra, char letra, int *tamanho) {
    int contador = 0;
    int i;

    for (i = 0; palavra[i] != '\0'; i++) {
        if (palavra[i] == letra) {
            contador++;
        }
    }

    int *posicao = (int *)malloc(contador * sizeof(int));

    if (posicao == NULL) {
        *tamanho = 0;
        return NULL;
    }

    int index = 0;
    for (i = 0; palavra[i] != '\0'; i++) {
        if (palavra[i] == letra) {
            posicao[index++] = i;
        }
    }

    *tamanho = contador;
    return posicao;
}
```

```
int main() {
    char palavra[50];
    char letra;
    int tamanho, i;

    printf("Digite uma palavra: ");
    scanf("%s", palavra);

    printf("Digite a letra que deseja pesquisar: ");
    scanf(" %c", &letra);

    int *posicao = acharPosicao(palavra, letra, &tamanho);
```

```

if (posicao != NULL) {
    printf("A letra '%c' aparece nas posicoes: ", letra);
    for (i = 0; i < tamanho; i++) {
        printf("%d ", posicao[i]);
    }
    printf("\nTotal de vezes que a letra aparece: %d\n", tamanho);

    free(posicao);
} else {
    printf("Erro em alocar memoria ou nao tem letras\n");
}

return 0;
}

```

12) O que significa o operador asterisco em cada um dos seguintes casos (cout tem como finalidade imprimir na tela):

- a) int *p; // Declaração de ponteiro
- b) cout << *p; // Imprimir o valor que o ponteiro recebe (o valor, não o endereço)
- c) *p = x*5; //O valor do ponteiro será x multiplicado por 5
- d) cout << *(p+1); //Imprimir o valor do ponteiro + 1

13) Qual e a saída deste programa?

```

#include <iostream.h>
void main(){
    int i=5, *p;
    p = &i; // *p = 5
    cout << p << '\t' << (*p+2) << '\t' << **&p << '\t' << (3**p) << '\t' << (**&p+4);
}

```

R:

p = endereço de i.

($*p + 2$) = 7

($**\&p$) = 5

($3 * *p$) = 15

($**\&p + 4$) = 9

STRUCTS

- 1) Escrever um programa que cadastre o nome, a matrícula e duas notas de varios alunos. Em seguida imprima a matricula, o nome e a média de cada um deles (utilizar funções).

R:

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

```
typedef struct {
    char nome[50];
    int matricula;
    float n1, n2;
    float media;
} Infoalunos;
```

```
void imprimirInfo(Infoalunos info[], int quantidade);
```

```
int main() {
    int quantidade, i;
    printf("Digite a quantidade de alunos para cadastrar: ");
    scanf("%d", &quantidade);

    Infoalunos info[quantidade];
    for (i = 0; i < quantidade; i++) {
        fflush(stdin);
        printf("\nDigite o nome da pessoa %d: \n", i+1);
        scanf("%s", info[i].nome);

        fflush(stdin);

        printf("Digite o numero da matricula: \n");
        scanf("%d", &info[i].matricula);

        fflush(stdin);

        printf("Digite a nota 1 da pessoa %d: \n", i+1);
        scanf("%f", &info[i].n1);

        printf("Digite a nota 2 da pessoa %d: \n", i+1);
        scanf("%f", &info[i].n2);

        info[i].media = (info[i].n1 + info[i].n2)/2;
    }

    imprimirInfo(info, quantidade);

    return 0;
}
```



```

void imprimirInfo(Infoalunos info[], int quantidade) {
    int i;
    for(i = 0; i < quantidade; i++) {
        printf("\nO nome da pessoa eh: %s\n", info[i].nome);
        printf("O numero da matricula da pessoa eh: %d\n", info[i].matricula);
        printf("A media dela eh: %f\n", info[i].media);
    }
}

```

- 2) Escrever um programa que cadastre o nome, a altura, o peso, o cpf e sexo de algumas pessoas. Com os dados cadastrados, em seguida localizar uma pessoas por meio do seu CPF e imprimir o seu IMC (utilizar funções).

R:

```

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

```

```

typedef struct {
    char nome[50];
    float altura, peso;
    int cpf;
    char sexo;
} Infopessoas;

```

```

void imprimirInfo(Infopessoas info[], int quantidade);

```

```

int main() {
    int quantidade, i;
    printf("Digite a quantidade de pessoas para cadastrar: ");
    scanf("%d", &quantidade);

    Infopessoas info[quantidade];
    for (i = 0; i < quantidade; i++) {
        fflush(stdin);
        printf("\nDigite o nome da pessoa %d: \n", i+1);
        scanf("%s", info[i].nome);

        fflush(stdin);

        printf("Digite a altura: \n");
        scanf("%f", &info[i].altura);
        printf("Digite o peso: \n");
        scanf("%f", &info[i].peso);

        fflush(stdin);

        printf("Digite o cpf da pessoa %d: \n", i+1);
        scanf("%d", &info[i].cpf);
    }
}

```

```

        fflush(stdin);
        printf("Digite o sexo da pessoa %d ('m' para masculino e 'f' para feminino):
\n", i+1);
        scanf("%c", &info[i].sexo);
    }

    imprimirInfo(info, quantidade);

    return 0;
}

void imprimirInfo(Infopessoas info[], int quantidade) {
    int i;
    for(i = 0; i < quantidade; i++) {
        printf("\nO nome da pessoa eh: %s\n", info[i].nome);
        printf("O peso da pessoa eh: %f\n", info[i].altura);
        printf("A altura da pessoa eh: %f\n", info[i].peso);
        printf("O CPF da pessoa eh: %d\n", info[i].cpf);
        printf("O sexo da pessoa eh: %c\n", info[i].sexo);
    }
}

```