**Lista de Exercícios**

1. Faça um algoritmo que leia o tamanho de um vetor de inteiros e reserve dinamicamente memória para esse vetor. Em seguida, leia os elementos desse vetor, imprima o vetor lido e mostre o resultado da soma dos números ímpares presentes no vetor.
2. Crie um algoritmo para preencher e imprimir os elementos de vetor com um certo valor. Para varrer o vetor, **não utilize índices, use apenas aritmética de ponteiros.**Faça isto para os vetores de inteiros e de reais**.**
3. Qual o efeito do seguinte fragmento de código?

char \*s;

s = "ABC";

printf ("%s\n", s);

1. O que há de errado com a seguinte variante do exercício anterior?

char s[20];

s = "ABC";

printf ("%s\n", s);

1. Analise o código abaixo e determine o valor das variáveis ao final da execução.

int i=34,j;

int \*p;

p = &i;

\*P++;

j = \*p + 33;

1. Analise o código abaixo e determine o valor das variáveis ao final da execução.

int i=7, j=5;

int \*p;

int \*\*q;

p = &i;

q = &p;

c = \*\*q + j;

1. Analise o código abaixo e determine o valor das variáveis ao final da execução.

int a=5, b=12, c=5;

int \*p;

int \*q;

p = &a;

q = &b;

(\*pe)++;

c = \*p + \*q;

1. O que faz o código abaixo?

#define Y 2

#define X 2

int main(int argc, char \*argv[])

{

int \*\*A; int i,j;

A = (int\*\*)malloc(Y \* sizeof(int\*));

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

A[i] = (int\*)malloc(X \* sizeof(int));

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

for(j = 0; j < X; j++)

A[i][j]=i+j;

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

for(j = 0; j < X; j++)

printf(%d \t”,A[i][j]);

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

free(A[i]);

free(A);

return 0;

}

1. Qual a diferença entre "mno" e "m\no"?  Qual a diferença entre "MNOP" e "MN0P"?  Qual a diferença entre "MN\0P" e "MN0P"?
2. Escreva um algoritmo que receba um caractere c e devolva uma string cujo único caractere é c.
3. Vazio versus NULL.  Suponha que str é uma string (ou seja, uma variável do tipo char \*).  Qual a diferença entre as afirmações str é uma string vazia e str é NULL?
4. O seguinte fragmento de código pretende decidir se abacate vem antes ou depois de banana no dicionário. O que há de errado?

char \*b, \*a;

a = "abacate";

b = "banana";

if (a < b) printf ("%s vem antes de %s\n", a, b);

else printf ("%s vem depois de %s\n", a, b);

1. O seguinte fragmento de código pretende decidir se abacate vem antes ou depois de amora no dicionário. O que há de errado?

char \*a, \*b;

a = "abacate";

b = "amora";

if (\*a < \*b) printf ("%s vem antes de %s\n", a, b);

else printf ("%s vem depois de %s\n", a, b);

1. Qual a diferença entre as expressões  **strcpy (s, t);**  e  **s = t ;** ?
2. Qual a diferença entre as expressões  if (strcmp (s, t) < 0)  e  if (s < t) ?
3. Discuta as diferenças entre os três fragmentos de código a seguir:

char a[8], b[8];

strcpy (a, "abacate");

strcpy (b, "banana");

/\* ---------------------------------------------------- \*/

char \*a, \*b;

a = (char\*) malloc (8);

strcpy (a, "abacate");

b = (char\*) malloc (8);

strcpy (b, "banana");

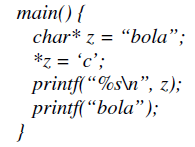
/\* ---------------------------------------------------- \*/

char \*a, \*b;

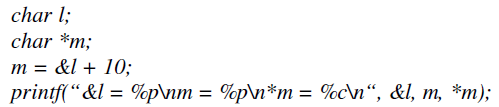
a = "abacate";

b = "banana";

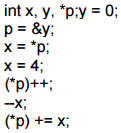
1. Edite o programa seguinte, compile-o e o execute. Relate o que ocorreu na compilação e durante a execução.



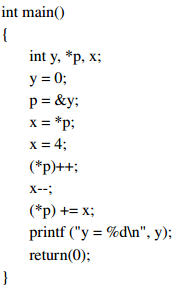
1. Edite o programa seguinte, compile-o e o execute. Relate o que ocorreu na compilação e durante a execução.



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



1. Explique a diferença entre
2. p++;
3. (\*p)++;
4. \*(p++);
5. Qual o valor de y no final do programa?

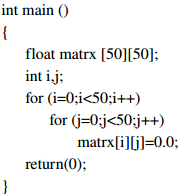


1. Diga quais expressões abaixo são válidas ou não. Considere as declarações

int vetor[10];

int \*ponteiro;

1. vetor = vetor + 2;
2. vetor++;
3. vetor = ponteiro;
4. ponteiro = vetor;
5. ponteiro = vetor+2;
6. Reescreva o programa abaixo usando ponteiros



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

int x, y, \*p,\*\*pp;

y = 0;

p = &y;

x = \*p;

x = 4;

pp = &p;

(\*p)++;

(\*\*pp)++;

--x;

(\*p) += x;

(\*\*pp)+=x;

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

**a)**

int 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)**

char \*a, \*b;

a = "abacate";

b = "uva";

if (a < b)

printf ("%s vem antes de %s no dicionário", a, b);

else

printf ("%s vem depois de %s no dicionário", a, b);

1. 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?
2. Suponha que v é um vetor. Descreva a diferença conceitual entre as expressões v[3] e v + 3.
3. Qual serão as saídas do seguinte programa?

|  |
| --- |
| #include <stdio.h>  int main() {  int valor;  int \*p1;  float temp;  float \*p2;  char aux;  char \*nome = "Algoritmos";  char \*p3;  int idade;  int vetor[3];  int \*p4;  int \*p5;    /\* (a) \*/  valor = 10;  p1 = &valor;  \*p1 = 20;  printf("(a) %d \n", valor);    /\* (b) \*/  temp = 26.5;  p2 = &temp;  \*p2 = 29.0;  printf("(b) %.1f \n", temp);    /\* (c) \*/  p3 = &nome[0];  aux = \*p3;  printf("(c) %c \n", aux);    /\* (d) \*/  p3 = &nome[4];  aux = \*p3;  printf("(d) %c \n", aux);    /\* (e) \*/  p3 = nome;  printf("(e) %c \n", \*p3);    /\* (f) \*/  p3 = p3 + 4;  printf("(f) %c \n", \*p3);    /\* (g) \*/  p3--;  printf("(g) %c \n", \*p3);    /\* <h> \*/  vetor[0] = 31;  vetor[1] = 45;  vetor[2] = 27;  p4 = vetor;  idade = \*p4;  printf("(h) %d \n", idade);    /\* (i) \*/  p5 = p4 + 1;  idade = \*p5;  printf("(i) %d \n", idade);    /\* (j) \*/  p4 = p5 + 1;  idade = \*p4;  printf("(j) %d \n", idade);    /\* (l) \*/  p4 = p4 - 2;  idade = \*p4;  printf("(l) %d \n", idade);    /\* (m) \*/  p5 = &vetor[2] - 1;  printf("(m) %d \n", \*p5);    /\* (n) \*/  p5++;  printf("(n) %d \n", \*p5);      return 0 ;  } |

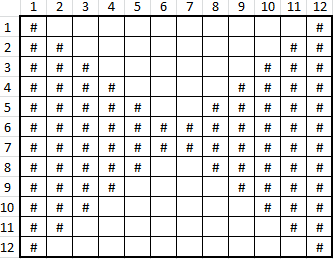
1. O que fazem os seguintes programas?

|  |  |  |
| --- | --- | --- |
| #include <stdio.h>  void main(){  int vet[] = {4,9,13};  int i;  for(i=0;i<3;i++)  {  printf("%d",\*(vet+i));  }  } | #include <stdio.h>  void main(){  int vet[] ={4,9,13};  int i;  for(i=0;i<3;i++){  printf("%X ",vet+i);  }  } | #include <stdio.h>  void main(){  int vet[] ={4,9,13};  int i;  for(i=0;i<3;i++){  printf("%X ",vet+i);  }  } |

1. Implemente um algoritmo que faça:
2. Alocação dinâmica para armazenar um valor inteiro.
3. Alocação dinâmica para armazenar um conjunto de 10 valores reais.
4. Alocação dinâmica para armazenar uma string de no máximo 49 caracteres.
5. Alocação dinâmica para armazenar uma lista de 15 nomes de no máximo 20 caracteres.

Para todas as alocações aplicadas efetue a liberação de memória de cada estrutura.

1. Faça um programa que acha o maior e o menor inteiro dentro de um vetor de 10 inteiros, calcule a soma e imprime a soma. Obs: usar apenas as variáveis a seguir: int v[10], i, \*maior, \*menor.
2. Escreva um programa em linguagem C que solicita ao usuário a quantidade de alunos de uma turma e aloca um vetor de notas (números reais). Depois de ler as notas, imprime a média aritmética. Obs: não deve ocorrer desperdício de memória; e após ser utilizada a memória deve ser devolvida.
3. Desenvolva um programa que calcule a soma de duas matrizes MxN de números reais (double). A implementação deste programa deve considerar as dimensões fornecida pelo usuário (Dica: represente a matriz através de variáveis do tipo double \*\*, usando alocação dinâmica de memória).
4. Faça um programa que leia um valor n e crie dinamicamente um vetor de n elementos, leia os elementos desse vetor. Depois, o vetor preenchido deve ser impresso. Além disso, antes de finalizar o programa, deve-se liberar a área de memória alocada.
5. Faça um algoritmo que crie uma matriz NxN, leia valores inteiros e retorne a soma da área hachurada da figura abaixo:



1. Explique a diferença entre (p é um ponteiro)
2. p++;
3. (\*p)++;
4. \*(p++);
5. Diga quais expressões abaixo são válidas ou não. Considere as declarações

int vetor[10];

int \*ponteiro;

1. vetor = vetor + 2;
2. vetor++;
3. vetor = ponteiro;
4. ponteiro = vetor;
5. ponteiro = vetor+2;
6. Quais serão os valores de x, y e p ao final do trecho de código abaixo?

int x, y, \*p,\*\*pp;

y = 0;

p = &y;

x = \*p;

x = 4;

pp = &p;

(\*p)++;

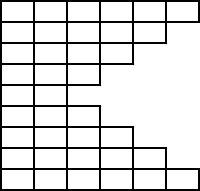
(\*\*pp)++;

--x;

(\*p) += x;

(\*\*pp)+=x;

1. Suponha que para resolver um problema matemático seja necessário criar uma matriz dinamicamente com o seguinte formato. Crie um algoritmo que aloque dinamicamente essa matriz.



1. O programa a seguir está correto? Se não, quais os comandos inválidos?

