

## Problema 1.

```
1  #include <stdio.h>
2
3  int main(){
4      int a[] = {1, 0, 5, -2, -5, 7};
5      int soma = a[0] + a[1] + a[5];
6      printf("%d\n", soma);
7      a[4] = 100;
8      for(int i=0; i<6; i++) printf("%d\n", a[i]);
9      return 0;
10 }
```

PROBLEMAS SAÍDA CONSOLE DE DEPURAÇÃO TERMINAL

```
igor@igor-Aspire-A315-56:~/Área de Trabalho/repositorio_faculdade/faculdade/lab_aeds1/lista_8_aeds$ gcc p1.c -o p1
igor@igor-Aspire-A315-56:~/Área de Trabalho/repositorio_faculdade/faculdade/lab_aeds1/lista_8_aeds$ ./p1
8
1
0
5
-2
100
7
igor@igor-Aspire-A315-56:~/Área de Trabalho/repositorio_faculdade/faculdade/lab_aeds1/lista_8_aeds$
```

## Problema 2.

```
1  #include <stdio.h>
2
3  void imprime_inteiro(){
4      int tam;
5      printf("Informe o tamanho do vetor: ");
6      scanf("%d", &tam);
7      int vetor[tam];
8      printf("Informe o valor para a posição 0: ");
9      scanf("%d", &vetor[0]);
10     int maior = vetor[0], posicao = 0;
11     for(int i = 1; i<tam; i++){
12         printf("Informe o valor para a posição %d: ", i);
13         scanf("%d", &vetor[i]);
14         if(vetor[i]>maior){
15             maior = vetor[i];
16             posicao = i;
17         }
18     }
19     printf("Maior: %d\nPosição: %d\n", maior, posicao);
20 }
21
22 int main(){
23     imprime_inteiro();
24     return 0;
25 }
```

```
igor@igor-Aspire-A315-56:~/Área de Trabalho/repositorio_faculdade/faculdade/lab_aeds1/lista_8_aeds$ gcc p2.c -o p2
igor@igor-Aspire-A315-56:~/Área de Trabalho/repositorio_faculdade/faculdade/lab_aeds1/lista_8_aeds$ ./p2
Informe o tamanho do vetor: 5
Informe o valor para a posição 0: 1
Informe o valor para a posição 1: 5
Informe o valor para a posição 2: 2
Informe o valor para a posição 3: 6
Informe o valor para a posição 4: 3
Maior: 6
Posição: 3
igor@igor-Aspire-A315-56:~/Área de Trabalho/repositorio_faculdade/faculdade/lab_aeds1/lista_8_aeds$
```

### Problema 3.

```
3 void zera_negativos(){
4     int tam;
5     printf("Informe o tamanho do vetor: ");
6     scanf("%d", &tam);
7     int vetor[tam];
8     for(int i = 0; i<tam; i++){
9         printf("Informe o valor para a posição %d: ", i);
10        scanf("%d", &vetor[i]);
11        if(vetor[i]<0){
12            vetor[i] = 0;
13        }
14    }
15    for(int i = 0; i<tam-1; i++) printf("%d, ", vetor[i]);
16    printf("%d.\n", vetor[tam-1]);
17 }
18
19 int main(){
20     zera_negativos();
21     return 0;
22 }
```

PROBLEMAS SAÍDA CONSOLE DE DEPURAÇÃO TERMINAL

```
● igor@igor-Aspire-A315-56:~/Área de Trabalho/repositorio_faculdade/faculdade/lab_aeds1/lista_8_aeds$ gcc p3.c -o p3
● igor@igor-Aspire-A315-56:~/Área de Trabalho/repositorio_faculdade/faculdade/lab_aeds1/lista_8_aeds$ ./p3
Informe o tamanho do vetor: 5
Informe o valor para a posição 0: 1
Informe o valor para a posição 1: -2
Informe o valor para a posição 2: 3
Informe o valor para a posição 3: -4
Informe o valor para a posição 4: 5
1, 0, 3, 0, 5.
○ igor@igor-Aspire-A315-56:~/Área de Trabalho/repositorio_faculdade/faculdade/lab_aeds1/lista_8_aeds$
```

### Problema 4.

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int verifica_vetor(int n, int *v, int tam){
5      for(int i = 0; i<tam; i++){
6          if(n == v[i]) return 1;
7      }
8      return 0;
9  }
10
11  int *vetor_sem_repeticao(int *vetor, int t){
12      int *novo_vetor = (int*)malloc(t * sizeof(int));
13      int j = 0;
14      for(int i = 0; i < t; i++){
15          if(!verifica_vetor(vetor[i], novo_vetor, t)){
16              novo_vetor[j] = vetor[i];
17              j++;
18          }
19      }
20      return novo_vetor;
21  }
22
23  int main(){
24      int vetor[] = {1, 2, 3, 4, 5, 6, 6, 6, 7, 8, 10, 1, 2};
25      int *x = vetor_sem_repeticao(vetor, 13);
26      for(int i = 0; i<13; i++){
27          printf("%d, ", x[i]);
28      }
29      printf("\n");
30      free(x);
31      return 0;
32  }
```

```

● igor@igor-Aspire-A315-56:~/Área de Trabalho/repositorio_faculdade/faculdade/lab_aeds1/lista_8_aeds$ gcc p4.c -o p4
● igor@igor-Aspire-A315-56:~/Área de Trabalho/repositorio_faculdade/faculdade/lab_aeds1/lista_8_aeds$ ./p4
1, 2, 3, 4, 5, 6, 7, 8, 10, 0, 0, 0, 0, 0,
○ igor@igor-Aspire-A315-56:~/Área de Trabalho/repositorio_faculdade/faculdade/lab_aeds1/lista_8_aeds$ █

```

### Problema 5.

```

1  #include <stdio.h>
2
3  int main(){
4      int n;
5      printf("Informe o tamanho dos vetores: ");
6      scanf("%d", &n);
7      double v1[n], v2[n], produto = 0;
8      for(int i=0; i<n; i++){
9          printf("Informe o %dº valor do vetor 1: ", i+1);
10         scanf("%lf", &v1[i]);
11         printf("Informe o %dº valor do vetor 2: ", i+1);
12         scanf("%lf", &v2[i]);
13         produto += v1[i]*v2[i];
14     }
15     printf("Produto escalar dos vetores: %.1lf\n", produto);
16     return 0;
17 }

```

```

● igor@igor-Aspire-A315-56:~/Área de Trabalho/repositorio_faculdade/faculdade/lab_aeds1/lista_8_aeds$ gcc p5.c -o p5
● igor@igor-Aspire-A315-56:~/Área de Trabalho/repositorio_faculdade/faculdade/lab_aeds1/lista_8_aeds$ ./p5
Informe o tamanho dos vetores: 3
Informe o 1º valor do vetor 1: 1
Informe o 1º valor do vetor 2: 2
Informe o 2º valor do vetor 1: 3
Informe o 2º valor do vetor 2: 4
Informe o 3º valor do vetor 1: 5
Informe o 3º valor do vetor 2: 6
Produto escalar dos vetores: 44.0
○ igor@igor-Aspire-A315-56:~/Área de Trabalho/repositorio_faculdade/faculdade/lab_aeds1/lista_8_aeds$ █

```

### Problema 6.

```

1  #include <stdio.h>
2
3  int main(){
4      char s[50];
5      printf("Digite uma string: ");
6      scanf("%s", s);
7      for(int i=0; i<4; i++){
8          printf("%c", s[i]);
9      }
10     printf("\n");
11     return 0;
12 }

```

```

● igor@igor-Aspire-A315-56:~/Área de Trabalho/repositorio_faculdade/faculdade/lab_aeds1/lista_8_aeds$ gcc p6.c -o p6
● igor@igor-Aspire-A315-56:~/Área de Trabalho/repositorio_faculdade/faculdade/lab_aeds1/lista_8_aeds$ ./p6
Digite uma string: abcdefgh
abcd
○ igor@igor-Aspire-A315-56:~/Área de Trabalho/repositorio_faculdade/faculdade/lab_aeds1/lista_8_aeds$ █

```

## Problema 7.

```
1  #include <stdio.h>
2
3  int main(){
4      char string1[50], string2[50];
5      scanf("%s %s", string1, string2);
6      for(int i=0; i<50; i++){
7          if(string1[i]<string2[i]){
8              printf("%s\n%s\n", string1, string2);
9              break;
10         }else if(string1[i]>string2[i]){
11             printf("%s\n%s\n", string2, string1);
12             break;
13         }
14     }
15 }
```

```
igor@igor-Aspire-A315-56:~/Área de Trabalho/repositorio_faculdade/faculdade/lab_aeds1/lista_8_aeds$ gcc p7.c -o p7
igor@igor-Aspire-A315-56:~/Área de Trabalho/repositorio_faculdade/faculdade/lab_aeds1/lista_8_aeds$ ./p7
estrutura
algoritmo
algoritmo
estrutura
igor@igor-Aspire-A315-56:~/Área de Trabalho/repositorio_faculdade/faculdade/lab_aeds1/lista_8_aeds$
```

## Problema 8.

```
1  #include <stdio.h>
2  #include <string.h>
3
4  int search(char *comparacao, char *fonte){
5      int cont = 0;
6      char *ocorrencia = strstr(fonte, comparacao);
7      while(ocorrencia != NULL){
8          cont++;
9          ocorrencia += strlen(comparacao);
10         ocorrencia = strstr(ocorrencia, comparacao);
11     }
12     return cont;
13 }
14
15 int main(){
16     char s1[50], s2[50];
17     scanf("%s %s", s1, s2);
18     printf("%d\n", search(s1, s2));
19     return 0;
20 }
```

```
igor@igor-Aspire-A315-56:~/Área de Trabalho/repositorio_faculdade/faculdade/lab_aeds1/lista_8_aeds$ gcc p8.c -o p8
igor@igor-Aspire-A315-56:~/Área de Trabalho/repositorio_faculdade/faculdade/lab_aeds1/lista_8_aeds$ ./p8
abc
abcdabcbabcbcd
3
igor@igor-Aspire-A315-56:~/Área de Trabalho/repositorio_faculdade/faculdade/lab_aeds1/lista_8_aeds$
```

## Problema 9.

```
1  #include <stdio.h>
2  #include <string.h>
3
4  int conta_vogais(char *string){
5      int cont = 0;
6      char vogais[] = "aeiouAEIOU";
7      char *ocorrencia = strpbrk(string, vogais);
8      while(ocorrencia != NULL){
9          cont++;
10         ocorrencia++;
11         ocorrencia = strpbrk(ocorrencia, vogais);
12     }
13     return cont;
14 }
15
16 char *troca_vogais(char *string, char c){
17     char vogais[] = "aeiouAEIOU";
18     char *ocorrencia = strpbrk(string, vogais);
19     while(ocorrencia != NULL){
20         *ocorrencia = c;
21         ocorrencia++;
22         ocorrencia = strpbrk(ocorrencia, vogais);
23     }
24     return string;
25 }
26
27 int main(){
28     char string[50];
29     printf("Digite uma frase com até 50 caracteres: ");
30     fgets(string, 50, stdin);
31     char c;
32     printf("Digite o caracter para substituir as vogais: ");
33     scanf("%c", &c);
34     printf("Foram encontradas %d vogais.\n", conta_vogais(string));
35     printf("%s", troca_vogais(string, c));
36     return 0;
37 }
```

```
• igor@igor-Aspire-A315-56:~/Área de Trabalho/repositorio_faculdade/faculdade/lab_aeds1/lista_8_aeds$ gcc p9.c -o p9
• igor@igor-Aspire-A315-56:~/Área de Trabalho/repositorio_faculdade/faculdade/lab_aeds1/lista_8_aeds$ ./p9
Digite uma frase com até 50 caracteres: Igor Augusto de Serpa e Cunha
Digite o caracter para substituir as vogais: X
Foram encontradas 12 vogais.
XgXr XXgXstX dX SXrpX X CXnhX
• igor@igor-Aspire-A315-56:~/Área de Trabalho/repositorio_faculdade/faculdade/lab_aeds1/lista_8_aeds$
```

## Problema 10.

```
1  #include <stdio.h>
2
3  int main(){
4      char string[50];
5      int i, j;
6      printf("Digite a string: ");
7      fgets(string, 50, stdin);
8      printf("Digite dois números: ");
9      scanf("%d %d", &i, &j);
10     for(; i<=j; i++) printf("%c", string[i]);
11     printf("\n");
12     return 0;
13 }
```

```
igor@igor-Aspire-A315-56:~/Área de Trabalho/repositorio_faculdade/faculdade/lab_aeds1/lista_8_aeds$ gcc p10.c -o p10
igor@igor-Aspire-A315-56:~/Área de Trabalho/repositorio_faculdade/faculdade/lab_aeds1/lista_8_aeds$ ./p10
Digite a string: Algoritmos e estrutura de dados
Digite dois números: 2 15
goritmos e est
igor@igor-Aspire-A315-56:~/Área de Trabalho/repositorio_faculdade/faculdade/lab_aeds1/lista_8_aeds$
```