

01 - sequência de fibonacci

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

```
int fibonacci(int n) {  
    if (n == 0) {  
        return 0;  
    } else if (n == 1) {  
        return 1;  
    } else {  
        return fibonacci(n - 1) + fibonacci(n - 2);  
    }  
}
```

```
int main() {  
    int n;  
    scanf("%d", &n);  
    int resultado = fibonacci(n);  
    printf("O %d termo da sequencia de Fibonacci e %d.\n", n, resultado);  
    return 0;  
}
```

02 - conversão de base numérica

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

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

```
void reverse(char *str, int len) {  
    for (int i = 0; i < len / 2; i++) {  
        char temp = str[i];  
        str[i] = str[len - i - 1];  
        str[len - i - 1] = temp;  
    }  
}
```

```
char* decimal_para_binario(int n) {  
    static char buffer[33];  
    int i = 0;  
    if (n == 0) {  
        buffer[i++] = '0';  
    } else {  
        while (n > 0) {  
            buffer[i++] = (n % 2) + '0';  
            n /= 2;  
        }  
        buffer[i] = '\0';  
        reverse(buffer, i);  
        return buffer;  
    }  
}
```

```
char* decimal_para_octal(int n) {  
    static char buffer[12];  
    int i = 0;  
    if (n == 0) {  
        buffer[i++] = '0';  
    } else {  
        while (n > 0) {  
            buffer[i++] = (n % 8) + '0';  
            n /= 8;  
        }  
        buffer[i] = '\0';  
        reverse(buffer, i);  
        return buffer;  
    }  
}
```

```
char* decimal_para_hexadecimal(int n) {  
    static char buffer[9];  
    int i = 0;  
    if (n == 0) {
```

```

        buffer[i++] = '0';
    } else {
        while (n > 0) {
            int rem = n % 16;
            if (rem < 10) {
                buffer[i++] = rem + '0';
            } else {
                buffer[i++] = rem - 10 + 'A';
            }
            n /= 16;
        }
        buffer[i] = '\0';
        reverse(buffer, i);
        return buffer;
    }
}

int main() {
    char sistema[20];
    int numero;
    scanf("%s", sistema);
    scanf("%d", &numero);

    if (strcmp(sistema, "binario") == 0) {
        char *resultado = decimal_para_binario(numero);
        printf("O numero em binario eh: %s\n", resultado);
    } else if (strcmp(sistema, "hexadecimal") == 0) {
        char *resultado = decimal_para_hexadecimal(numero);
        printf("O numero em hexadecimal eh: %s\n", resultado);
    } else if (strcmp(sistema, "octal") == 0) {
        char *resultado = decimal_para_octal(numero);
        printf("O numero em octal eh: %s\n", resultado);
    } else {
        printf("Sistema de destino invalido!\n");
    }

    return 0;
}

```

03 - ordem de circuitos potentes

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

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

```
typedef struct {  
    char nome[31];  
    float tensao;  
    float corrente;  
    float potencia;  
    int indice;  
} Circuito;
```

```
int comparar_potencia(const void *a, const void *b) {  
    const Circuito *circA = (const Circuito *)a;  
    const Circuito *circB = (const Circuito *)b;  
  
    if (circB->potencia > circA->potencia)  
        return 1;  
    else if (circB->potencia < circA->potencia)  
        return -1;  
    else  
        return circA->indice - circB->indice;  
}
```

```
int main() {  
    int N;  
    scanf("%d", &N);  
  
    Circuito circuitos[N];  
    for (int i = 0; i < N; i++) {  
        scanf("%s", circuitos[i].nome);  
        scanf("%f", &circuitos[i].tensao);  
        scanf("%f", &circuitos[i].corrente);  
        circuitos[i].potencia = circuitos[i].tensao * circuitos[i].corrente;  
        circuitos[i].indice = i;  
    }  
  
    qsort(circuitos, N, sizeof(Circuito), comparar_potencia);  
  
    printf("\nCircuitos em ordem decrescente de potencia:\n");  
    for (int i = 0; i < N; i++) {  
        printf("%s - Potencia: %.2f W\n", circuitos[i].nome, circuitos[i].potencia);  
    }  
  
    return 0;  
}
```

04 - soma de dois vetores

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

```
int main() {
    int vetorA[5], vetorB[5];

    while (1) {
        // Lê os 5 elementos do primeiro vetor
        for (int i = 0; i < 5; i++) {
            if (scanf("%d", &vetorA[i]) != 1) {
                return 0; // Encerra se não houver entrada suficiente
            }
        }

        // Lê os 5 elementos do segundo vetor
        for (int i = 0; i < 5; i++) {
            if (scanf("%d", &vetorB[i]) != 1) {
                return 0;
            }
        }

        // Calcula a soma dos vetores
        int resultado[5];
        for (int i = 0; i < 5; i++) {
            resultado[i] = vetorA[i] + vetorB[i];
        }

        // Exibe o resultado formatado
        printf("Vetor resultante da soma: ");
        for (int i = 0; i < 5; i++) {
            printf("%d", resultado[i]);
            if (i < 4) {
                printf(" ");
            }
        }
        printf("\n");
    }

    return 0;
}
```

05 - senha forte senha fraca

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

int main() {
    char senha[100];
    scanf("%s", senha);

    int len = strlen(senha);
    int tem_minuscula = 0, tem_maiuscula = 0, tem_especial = 0;

    // Verifica cada caractere da senha
    for (int i = 0; i < len; i++) {
        char c = senha[i];
        if (islower(c)) {
            tem_minuscula = 1;
        } else if (isupper(c)) {
            tem_maiuscula = 1;
        } else if (strchr("!@#$%^&*()", c) != NULL) {
            tem_especial = 1;
        }
    }

    int erros = 0;

    // Verifica o comprimento da senha
    if (len < 8) {
        return printf("A senha deve ter pelo menos 8 caracteres.\n");
        erros++;
    }

    // Verifica se tem letra minúscula
    if (!tem_minuscula) {
        printf("A senha deve conter pelo menos uma letra minuscula.\n");
        erros++;
    }

    // Verifica se tem letra maiúscula
    if (!tem_maiuscula) {
        printf("A senha deve conter pelo menos uma letra maiuscula.\n");
        erros++;
    }

    // Verifica se tem caractere especial
    if (!tem_especial) {
        printf("A senha deve conter pelo menos um caractere especial(!@#$%^&*()).\n");
        erros++;
    }
}
```

```
}

// Se não houver erros, exibe que a senha é forte
if (erros == 0) {
    printf("Senha forte!\n");
}

return 0;
}
```

06 - transposição de matrizes

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

```
int main() {
    int n, m;
    scanf("%d %d", &n, &m);

    int original[n * m];
    int *ptr = original;
    for (int i = 0; i < n * m; i++) {
        scanf("%d", ptr++);
    }

    int transposed[m * n];
    for (int j = 0; j < m; j++) {
        for (int i = 0; i < n; i++) {
            *(transposed + j * n + i) = *(original + i * m + j);
        }
    }

    for (int j = 0; j < m; j++) {
        int *row_ptr = transposed + j * n;
        for (int i = 0; i < n; i++) {
            printf("%d ", *(row_ptr + i));
        }
        printf("\n");
    }

    return 0;
}
```


07 - valores críticos de um robô

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

```
int main() {
    int matriz[3][3];
    int *ptr = &matriz[0][0]; // Ponteiro para o primeiro elemento da matriz

    // Lê os 9 elementos da matriz
    for (int i = 0; i < 9; i++) {
        scanf("%d", ptr + i);
    }

    printf("Diagonal principal da matriz (valores críticos):\n");
    for (int i = 0; i < 3; i++) {
        int *elemento = ptr + i * 3 + i; // Calcula o endereço do elemento [i][i]
        printf("Elemento [%d][%d]: %d\n", i, i, *elemento);
    }

    return 0;
}
```

08 - tiro ao alvo

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

```
int calcular_pontos(int premio[3][3], int posicao[3][3], int tiros[3]) {
    int soma = 0;
    for (int k = 0; k < 3; k++) {
        int tiro = tiros[k];
        for (int i = 0; i < 3; i++) {
            for (int j = 0; j < 3; j++) {
                if (posicao[i][j] == tiro) {
                    soma += premio[i][j];
                    premio[i][j] = 0;
                    i = 3; // Força saída dos loops
                    j = 3;
                    break;
                }
            }
        }
    }
    return soma;
}
```

```
int main() {
    while (1) {
        int premio[3][3];
        int posicao[3][3] = {{1, 2, 3}, {4, 5, 6}, {7, 8, 9}};
        int tiros[3];
        int sum;
        double multiplicador;

        // Ler a matriz de prêmios
        int leitura_valida = 1;
        for (int i = 0; i < 3; i++) {
            for (int j = 0; j < 3; j++) {
                if (scanf("%d", &premio[i][j]) != 1) {
                    leitura_valida = 0;
                    i = 3; // Sair do loop externo
                    break;
                }
            }
        }
        if (!leitura_valida) break;

        // Ler os disparos
        for (int k = 0; k < 3; k++) {
            if (scanf("%d", &tiros[k]) != 1) {
                leitura_valida = 0;
                break;
            }
        }
    }
}
```

```

    }
    }
    if (!leitura_valida) break;

    sum = calcular_pontos(premio, posicao, tiros);

    printf("Pontos : %d\n", sum);

    // Imprimir a matriz atualizada
    for (int i = 0; i < 3; i++) {
        for (int j = 0; j < 3; j++) {
            printf(" %d ", premio[i][j]);
        }
        printf("\n");
    }

    // Calcular multiplicador
    multiplicador = 1.0;
    if (sum % 7 == 0) {
        multiplicador = 3.0;
    } else if (sum % 3 == 0) {
        multiplicador = 1.8;
    } else if (sum % 2 == 0) {
        multiplicador = 1.5;
    }

    double premio_final = sum * multiplicador;
    printf("Aqui estao premio em ticket que Joao teve : %.3f\n", premio_final);
}

return 0;
}

```