

Exercício de Programação Estruturada 13

Exercício de Programação Estruturada 13

1.

```
#include <stdio.h>
#include <ctype.h>
#define QUESTIONS 10
#define STUDENTS 3
#define NAMELEN 100

void fillSolution(char solution[QUESTIONS])
{
    int i;
    for (i = 0; i < QUESTIONS; i++)
    {
        printf("Informe a solução da questão %d: ", i + 1);
        scanf(" %c", &solution[i]);
    }
}

void fillAnswers(char answers[STUDENTS][QUESTIONS], char names[STUDENTS]
[NAMELEN])
{
    int i, j;
    for (i = 0; i < STUDENTS; i++)
    {
        printf("Informe o nome do aluno %d: ", i + 1);
        scanf(" %s", &names[i]);
        for (j = 0; j < QUESTIONS; j++)
        {
            printf("Informe a resposta de %s para a questão %d: ", names[i],
j + 1 );
            scanf(" %c", &answers[i][j]);
        }
    }
}
```

```

void correctAnswers(char answers[STUDENTS][QUESTIONS], char
solution[QUESTIONS], int correct[STUDENTS])
{
    int i, j, counter;
    for (i = 0; i < STUDENTS; i++)
    {
        counter = 0;
        for (j = 0; j < QUESTIONS; j++)
        {
            if (tolower(answers[i][j]) == tolower(solution[j]))
            {
                counter = 1 + counter;
            }
        }
        correct[i] = counter;
    }
}

void studentsSituation(char names[STUDENTS][NAMELEN], int correct[STUDENTS])
{
    int i;
    for (i = 0; i < STUDENTS; i++)
    {
        if (correct[i] >= (QUESTIONS/10) * 7)
        {
            printf("%d", correct[i]);
            printf("O aluno %s obteve nota %d e está aprovado.\n",
names[i], correct[i]);
        }
        else
        {
            printf("O aluno %s obteve nota %d e está reprovado.\n",
names[i], correct[i]);
        }
    }
}

int main() {
    char solution[QUESTIONS], answers[STUDENTS][QUESTIONS], names[STUDENTS]
[NAMELEN];
    int correct[STUDENTS];

```

```

    fillSolution(solution);
    fillAnswers(answers, names);
    correctAnswers(answers, solution, correct);
    studentsSituation(names, correct);
    return 0;
}

```

2.

```

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

int valida(char conta[], char dv) {
    int tam = strlen(conta), vet[tam], i, soma, resto, numDV, power;
    if (tam != 6) {
        return 0;
    } else {
        soma = 0;
        for (i = 0; i < tam; i++) {
            vet[i] = conta[i] - '0';
            power = (int) pow(2, i);
            soma = soma + vet[i] * power;
        }
        resto = soma % 11;
        if (resto >= 0 && resto <= 9) {
            numDV = dv - '0';
            if (numDV == resto) {
                return 1;
            } else {
                return 0;
            }
        } else {
            if (dv == 'x' || dv == 'X') {
                return 1;
            } else {
                return 0;
            }
        }
    }
}

int main() {
    char conta[7], dv;
    int retorno;
}

```

```
printf("Conta: ");
scanf(" %s", conta);
fflush(stdin);
printf("DV: ");
scanf(" %c", &dv);
fflush(stdin);
retorno = valida(conta, dv);
if (retorno == 0) {
    printf("Dígito verificado inválido!\n");
} else {
    printf("Dígito verificado válido!\n");
}
}
```