

Solução em C do Problema 8

```
/*
 * SOLUCAO 1: simples e direta. Note o cuidado especial com o
 *      ultimo segmento e ultimo elemento.
 *
 */

#include <stdio.h>

#define FALSE 0
#define TRUE 1

int main()
{
    int n;           /* no. de elementos na sequencia */
    int i;           /* contador de numeros lidos */
    int numero;       /* usado para ler a sequencia */
    int paridade;     /* indica a paridade do segmento (0 = par, 1 = impar) */
    int k;            /* candidato a grau da alternacia */
    int kaux;         /* tamanho do segmento corrente */
    int k_alternante; /* indica se a sequencia e' ou nao k-alternate */

    printf("Determino se uma sequencia de n (>0) inteiros e k-alternante.\n");

    /* leia o tamanho da sequencia */
    printf("Entre com n: ");
    scanf ("%d", &n);

    /* determina candidato a k e paridade do primeiro 1o. segmento */
    printf("Entre com o 1o. numero da sequencia: ");
    scanf ("%d", &numero);
    paridade = numero % 2;
    k = 1;
    i = 1;
    while (i < n && numero % 2 == paridade)
    {
        printf("Entre com o %do. numero da sequencia: ", i+1);
        scanf ("%d", &numero);
        i = i + 1; /* mais um numero foi lido */
        if (numero % 2 == paridade)
        {
            k = k + 1;
        }
    }

    /* neste ponto vale que k e' o comprimento do 1o. segmento de
     * mesma paridade.
     */
    printf("candidato a k = %d\n", k);
}
```

```

/* a sequencia e' k-alternante ate que se prove o contrario */
k_alternante = TRUE;
while (i < n)
{
    /* ajuste a alternancia da paridade */
    paridade = (paridade + 1) % 2;

    kaux = 1;
    while (i < n && numero % 2 == paridade)
    {
        printf("Entre com o %do. numero da sequencia: ", i+1);
        scanf ("%d", &numero);
        i = i + 1; /* mais um numero foi lido */
        if (numero % 2 == paridade)
        {
            kaux = kaux + 1;
        }
    }

    if (kaux != k)
    {
        k_alternante = FALSE;
    }
}

/* o ultimo elemento merece um cuidado especial... */
if (numero % 2 != paridade && k != 1)
{
    k_alternante = FALSE;
}

if (k_alternante == TRUE)
    printf("A sequencia e %d-alternante\n", k);
else
    printf("A sequencia nao e' k-alternante\n");

return 0;
}

/*
* SOLUCAO 2: similar a solucao 1, mas sem laco encaixado.
*
*/

#include <stdio.h>

#define FALSE 0
#define TRUE 1

int main()
{
    int n;           /* no. de elementos na sequencia */
    int i;           /* contador de numeros lidos */
    int numero;       /* usado para ler a sequencia */
    int paridade;     /* indica a paridade do segmento (0 = par, 1 = impar) */
    int k;            /* candidato a grau da alternancia */
    int kaux;         /* tamanho do segmento corrente */
    int k_alternante; /* indica se a sequencia e' ou nao k-alternate */

    printf("Determino se uma sequencia de n (>0) inteiros e k-alternante.\n");

```

```

/* leia o tamanho da sequencia */
printf("Entre com n: ");
scanf ("%d", &n);

/* leia o 1o. elemento e determine a paridade do 1o. segmento */
printf("Entre com o 1o. numero da sequencia: ");
scanf ("%d", &numero);
paridade = numero % 2;

/* determine o candidato a k */
k = 1;
i = 1;
while (i < n && numero % 2 == paridade)
{
    printf("Entre com o %do. numero da sequencia: ", i+1);
    scanf ("%d", &numero);
    i = i + 1; /* mais um numero foi lido */
    if (numero % 2 == paridade)
    {
        k = k + 1;
    }
}

printf("candidato a k = %d\n", k);
/* neste ponto vale que k e' o comprimento do 1o. segmento de
 * mesma paridade.
 */

/* a sequencia e' k-alternante ate que se prove o contrario */
k_alternante = TRUE;
kaux = 1; /* tamanho do segmento atual */
paridade = (paridade + 1) % 2; /* ajusta paridade */
while (i < n)
{
    /* ainda precisamos ler algum elemento */
    printf("Entre com o %do. numero da sequencia: ", i+1);
    scanf ("%d", &numero);
    i = i + 1; /* mais um elemento foi lido */

    if (numero % 2 == paridade)
    {
        /* estamos no mesmo segmento */
        kaux = kaux + 1;
    }
    else
    {
        /* ocorreu uma quebra de segmento */
        if (kaux != k)
        {
            k_alternante = FALSE;
        }
        kaux = 1;
        paridade = (paridade + 1) % 2;
    }
}

/* o ultimo segmento merece um cuidado especial... */
if (kaux != k)

```

```

    {
        k_alternante = FALSE;
    }

    if (k_alternante == TRUE)
        printf("A sequencia e %d-alternante\n", k);
    else
        printf("A sequencia nao e' k-alternante\n");

    return 0;
}

/*
 * SOLUCAO 3: solucao sem laco encaixado e com o tratamento do
 *          ultimo segmento dentro do laco principal.
 *
 */

#include <stdio.h>

#define FALSE 0
#define TRUE 1

int main()
{
    int n;           /* no. de elementos na sequencia */
    int i;           /* contador de numeros lidos */
    int numero;       /* usado para ler a sequencia */
    int paridade;     /* indica a paridade do segmento (0 = par, 1 = impar) */
    int k;            /* candidato a grau da alternancia */
    int kaux;         /* tamanho do segmento corrente */
    int k_alternante; /* indica se a sequencia e' ou nao k-alternate */

    printf("Determino se uma sequencia de n (>0) inteiros e k-alternante.\n");

    /* leia o tamanho da sequencia */
    printf("Entre com n: ");
    scanf ("%d", &n);

    /* leia o 1o. elemento e determine a paridade do 1o. segmento */
    printf("Entre com o 1o. numero da sequencia: ");
    scanf ("%d", &numero);
    paridade = numero % 2;

    /* determine o candidato a k */
    k = 1;
    i = 1;
    while (i < n && numero % 2 == paridade)
    {
        printf("Entre com o %do. numero da sequencia: ", i+1);
        scanf ("%d", &numero);
        i = i + 1; /* mais um numero foi lido */
        if (numero % 2 == paridade)
        {
            k = k + 1;
        }
    }

    printf("candidato a k = %d\n", k);
    /* neste ponto vale que k e' o comprimento do 1o. segmento de
     * mesma paridade.

```

```

*/

/* a sequencia e' k-alternante ate que se prove o contrario */
k_alternante = TRUE;
kaux = 1; /* tamanho do segmento atual */
paridade = (paridade + 1) % 2; /* ajusta paridade */
while (i <= n)
{
    if (i == n)
    {
        /* acabou de ler a sequencia */
        if (kaux != k)
        {
            k_alternante = FALSE;
        }
    }
    else
    {
        /* ainda precisamos ler algum elemento */
        printf("Entre com o %do. numero da sequencia: ", i+1);
        scanf ("%d", &numero);

        if (numero % 2 == paridade)
        {
            kaux = kaux + 1;
        }
        else
        {
            if (kaux != k)
            {
                k_alternante = FALSE;
            }
            kaux = 1;
            paridade = (paridade + 1) % 2;
        }
    }
    i = i + 1;
}

if (k_alternante == TRUE)
    printf("A sequencia e %d-alternante\n", k);
else
    printf("A sequencia nao e' k-alternante\n");

return 0;
}

/*
* SOLUCAO 4: eta solucao bonita.
*
*/

#include <stdio.h>

#define FALSE 0
#define TRUE 1

int main()
{
    int n;          /* no. de elementos na sequencia */

```

```

int i;                /* contador de numeros lidos */
int numero;           /* usado para ler a sequencia */
int paridade;         /* indica a paridade do segmento (0 = par, 1 = impar) */
int k;                /* candidato a grau da alternancia */
int kaux;             /* tamanho do segmento corrente */
int k_alternante;     /* indica se a sequencia e' ou nao k-alternante */

printf("Determino se uma sequencia de n (>0) inteiros e k-alternante.\n");

/* leia o tamanho da sequencia */
printf("Entre com n: ");
scanf ("%d", &n);

/* leia o 1o. elemento e determine a sua paridade */
printf("Entre com o 1o. numero da sequencia: ");
scanf ("%d", &numero);
paridade = numero % 2;
kaux = 1; /* comprimento corrente do segmento */

/* a sequencia e' k-alternante ate que se prove o contrario */
k_alternante = TRUE;
k = 0; /* o valor de k ainda precisa ser determinado */
for (i = 1; i <= n; i++)
{
    /* caso 1: i == n */
    if (i == n)
    {
        /* acabou de ler a sequencia */
        /* caso 1A: 1o. segmento */
        if (k == 0)
        {
            /* a sequencia e' composta por numeros de mesma paridade */
            k = kaux;
        }
        /* caso 1B: nao e' o 1o. segmento */
        else if (kaux != k)
        {
            k_alternante = FALSE;
        }
    }
    /* caso 2: i != n */
    else
    {
        printf("Entre com o %do. numero da sequencia: ", i+1);
        scanf ("%d", &numero);

        /* caso 2A: numero % 2 == paridade */
        if (numero % 2 == paridade)
        {
            kaux = kaux + 1;
        }
        /* caso 2B: numero % 2 != paridade */
        else
        {
            /* caso 2B1: 1o. segmento */
            if (k == 0)
            {
                k = kaux;
            }
            /* caso 2B2: nao e' o 1o. segmento */
            else if (kaux != k)

```

```

        {
            k_alternante = FALSE;
        }
        kaux = 1;
        paridade = (paridade + 1) % 2;
    }
}

if (k_alternante == TRUE)
    printf("A sequencia e %d-alternante\n", k);
else
    printf("A sequencia nao e' k-alternante\n");

return 0;
}

/*
 * SOLUCAO 5: solucao 4 sem chaves superfluas
 *
 */

#include <stdio.h>

#define FALSE 0
#define TRUE 1

int main()
{
    int n;           /* no. de elementos na sequencia */
    int i;           /* contador de numeros lidos */
    int numero;      /* usado para ler a sequencia */
    int paridade;    /* indica a paridade do segmento (0 = par, 1 = impar) */
    int k;           /* candidato a grau da alternancia */
    int kaux;        /* tamanho do segmento corrente */
    int k_alternante; /* indica se a sequencia e' ou nao k-alternate */

    printf("Determino se uma sequencia de n (>0) inteiros e k-alternante.\n");

    /* leia o tamanho da sequencia */
    printf("Entre com n: ");
    scanf ("%d", &n);

    /* leia o 1o. elemento e determine a sua paridade */
    printf("Entre com o 1o. numero da sequencia: ");
    scanf ("%d", &numero);
    paridade = numero % 2;
    kaux = 1; /* comprimento corrente do segmento */

    /* a sequencia e' k-alternante ate que se prove o contrario */
    k_alternante = TRUE;
    k = 0; /* o valor de k ainda precisa ser determinado */
    for (i = 1; i <= n; i++)
        if (i == n)
        {
            /* 1o. segmento */
            if (k == 0) k = kaux;
            /* nao e' o 1o. segmento */
            else if (kaux != k) k_alternante = FALSE;
        }
    else

```

```

{
    printf("Entre com o %do. numero da sequencia: ", i+1);
    scanf ("%d", &numero);

    if (numero % 2 == paridade) kaux++;
    else
    {
        /* 1o. segmento */
        if (k == 0) k = kaux;
        /* nao e' o 1o. segmento */
        else if (kaux != k) k_alternante = FALSE;

        kaux = 1;
        paridade = (paridade + 1) % 2;
    }
}

if (k_alternante == TRUE)
    printf("A sequencia e %d-alternante\n", k);
else
    printf("A sequencia nao e' k-alternante\n");

return 0;
}

```

