

Prima prova di esonero – 5 novembre 2010

La prova dura tre ore, durante le quali non è possibile allontanarsi dall'aula, se non dopo aver consegnato l'elaborato scritto. Per superare la prova di esonero e poter sostenere la successiva è necessario ottenere almeno 15 punti. È possibile utilizzare libri e appunti personali, senza scambiarli con altri studenti. I compiti che presenteranno evidenti ed anomale "similitudini" saranno annullati.

Esercizio n. 1 (max 10 punti)

Risolvere il seguente problema proponendo una pseudo-codifica dell'algoritmo, il diagramma di flusso ed infine la codifica in linguaggio C del programma che implementa l'algoritmo stesso.

Leggere in input una sequenza di n numeri interi positivi e memorizzarla nell'array A . Costruire un array B di n elementi, tali che

$$b_i = \sum_{k=0}^{\lfloor \frac{i}{2} \rfloor} a_k - \sum_{h=\lfloor \frac{i}{2} \rfloor + 1}^i a_h$$

dove si è indicato con $i/2$ la parte intera della divisione (es.: $3/2 = 1$). Stampare l'array B .

Esempio Supponiamo che vengano acquisiti in input $n = 6$ numeri memorizzati nel vettore $A = (6, 2, 4, 5, 1, 4)$. Allora il vettore B è il seguente: $B = (6, 4, 4, -1, 6, 2)$. Infatti:

$$\begin{aligned} b_0 &= a_0 = 6 \\ b_1 &= a_0 - a_1 = 6 - 2 = 4 \\ b_2 &= (a_0 + a_1) - a_2 = (6 + 2) - 4 = 4 \\ b_3 &= (a_0 + a_1) - (a_2 + a_3) = (6 + 2) - (4 + 5) = -1 \\ b_4 &= (a_0 + a_1 + a_2) - (a_3 + a_4) = (6 + 2 + 4) - (5 + 1) = 6 \\ b_5 &= (a_0 + a_1 + a_2) - (a_3 + a_4 + a_5) = (6 + 2 + 4) - (5 + 1 + 4) = 2 \end{aligned}$$

Soluzione

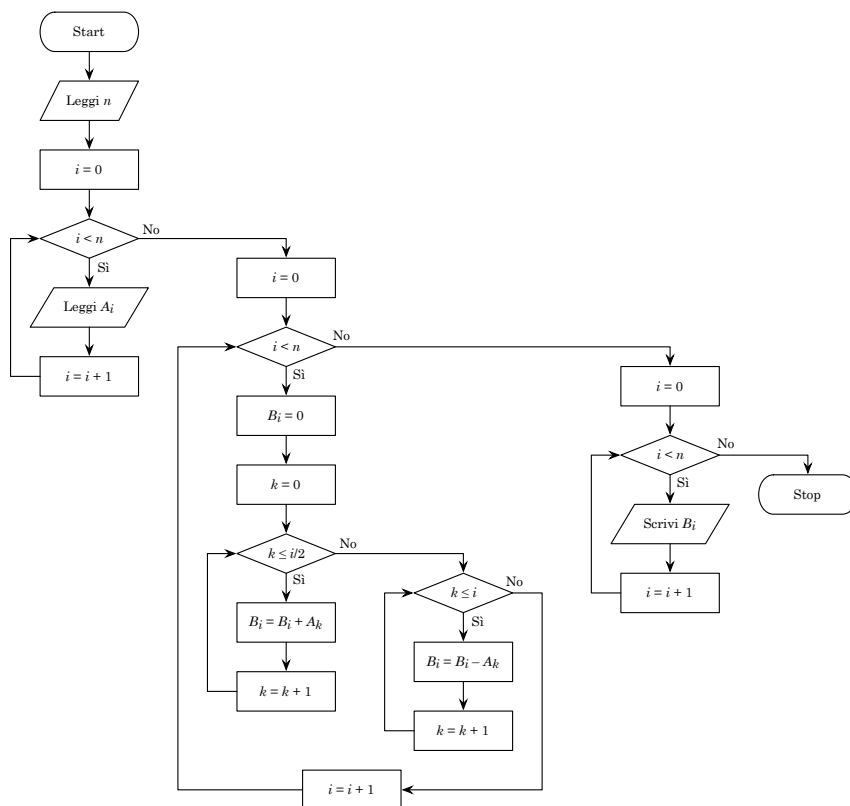
Pseudo-codifica dell'algoritmo

```

1: leggi  $n$ 
2: per  $i = 0, 1, 2, \dots, n - 1$  ripeti
3:   leggi  $a_i$ 
4: fine-ciclo
5: per  $i = 0, 1, 2, \dots, n - 1$  ripeti
6:    $b_i = 0$ 
7:   per  $k = 0, 1, 2, \dots, \lfloor i/2 \rfloor$  ripeti
8:      $b_i = b_i + a_k$ 
9:   fine-ciclo
10:  per  $k = \lfloor i/2 \rfloor + 1, \lfloor i/2 \rfloor + 2, \dots, i$  ripeti
11:     $b_i = b_i - a_k$ 
12:  fine-ciclo
13: fine-ciclo
14: per  $i = 0, 1, 2, \dots, n - 1$  ripeti
15:   scrivi  $b_i$ 
16: fine-ciclo

```

Diagramma di flusso



Codifica in linguaggio C

```

1 #include <stdlib.h>
2 #include <stdio.h>
3 #define MAX 100
4
5 int main(void) {
6     int n, A[MAX], B[MAX], k, i;
7     printf("Numero di elementi: ");
8     scanf("%d", &n);
9     printf("Inserisci %d numeri interi: ", n);
10    for (i=0; i<n; i++)
11        scanf("%d", &A[i]);
12    for (i=0; i<n; i++) {
13        B[i] = 0;
14        for (k=0; k<=i/2; k++)
15            B[i] = B[i]+A[k];
16        for (k=i/2+1; k<=i; k++)
17            B[i] = B[i]-A[k];
18    }
19    for (i=0; i<n; i++)
20        printf("%d ", B[i]);
21    return(0);
22 }
  
```

Esercizio n. 2 (max 8 punti)

Scrivere il diagramma di flusso e la codifica in linguaggio C (un programma completo) del seguente algoritmo:

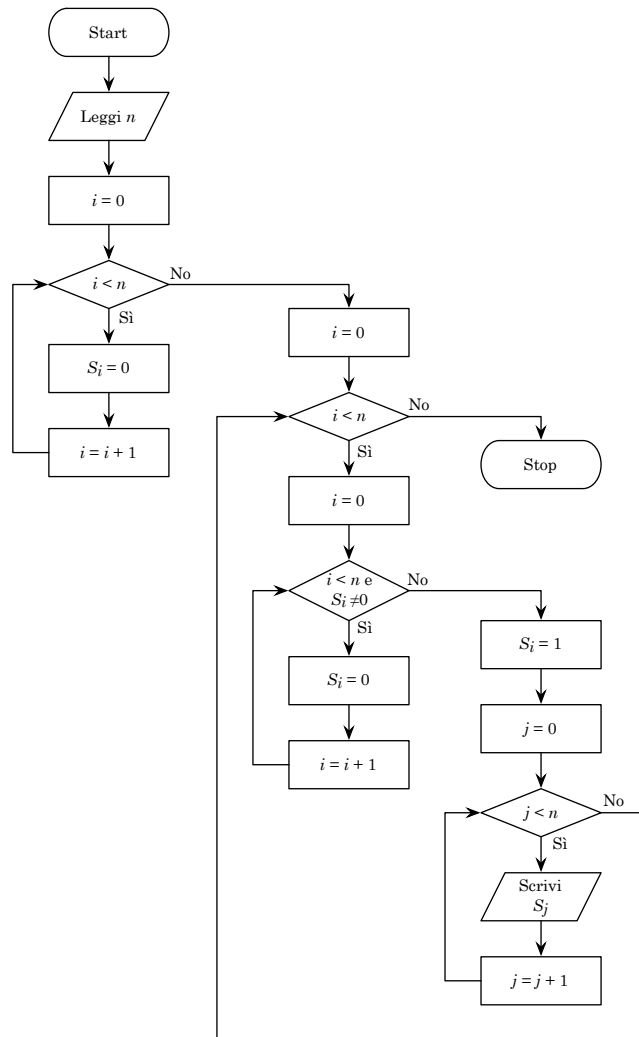
```
1: leggi  $n$ 
2:  $i = 0$ 
3: se  $i \geq n$  vai al passo 7
4:    $S_i = 0$ 
5:    $i = i + 1$ 
6: vai al passo 3
7:  $i = 0$ 
8: fin tanto che  $i < n$  ripeti
9:    $i = 0$ 
10:  fin tanto che  $i < n$  e  $S_i \neq 0$  ripeti
11:     $S_i = 0$ 
12:     $i = i + 1$ 
13:  fine-ciclo
14:   $S_i = 1$ 
15:  per  $j = 0, 1, 2, \dots, n - 1$  ripeti
16:    scrivi  $S_j$ 
17:  fine-ciclo
18: fine-ciclo
```

Soluzione

Codifica in linguaggio C

```
1 #include <stdlib.h>
2 #include <stdio.h>
3 #define MAX 50
4
5 int main(void) {
6     int i, j, n, S[MAX];
7     printf("Inserisci n: ");
8     scanf("%d", &n);
9     for (i=0; i<n; i++)
10        S[i] = 0;
11    i = 0;
12    while (i<n) {
13        for (i=0; i<n && S[i]!=0; i++)
14            S[i] = 0;
15        S[i] = 1;
16        for (j=0; j<n; j++)
17            printf("%d", S[j]);
18        printf("\n");
19    }
20    return(0);
21 }
```

Diagramma di flusso

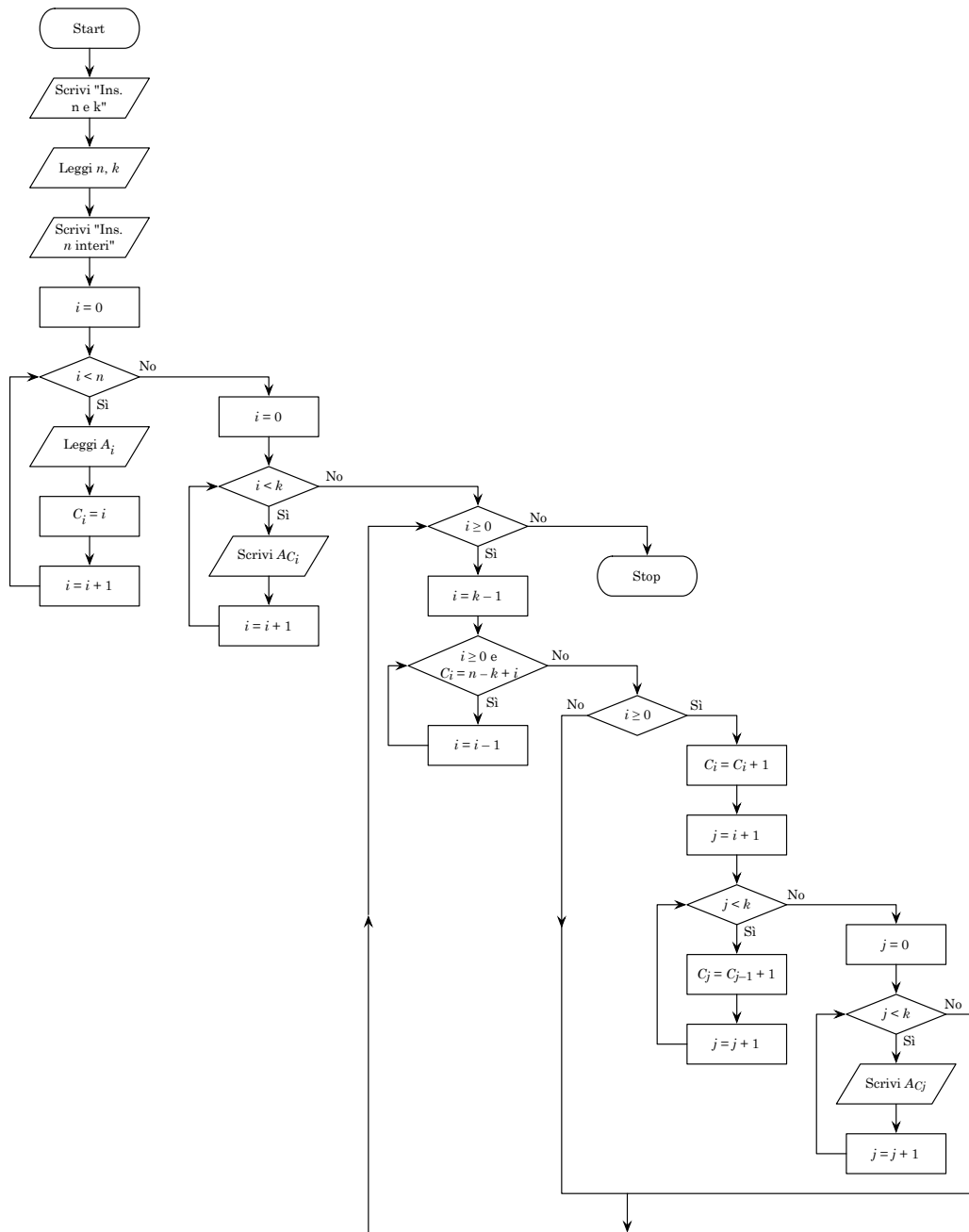


Esercizio n. 3 (max 6 punti)

Utilizzando le regole della programmazione strutturata, disegnare il diagramma di flusso della seguente funzione codificata in linguaggio C.

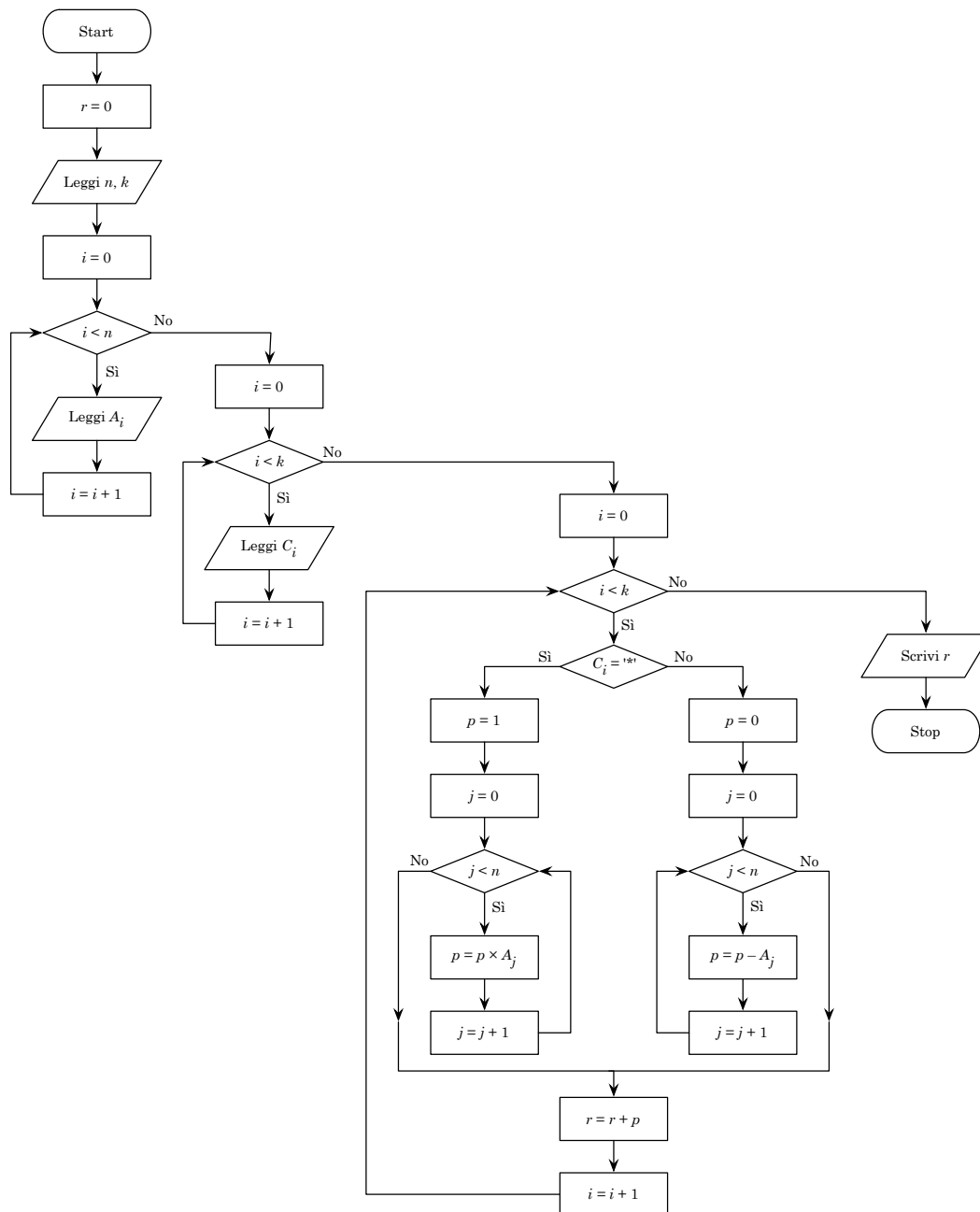
```
1 #include <stdlib.h>
2 #include <stdio.h>
3 #define MAX 100
4
5 int main(void) {
6     int A[MAX], C[MAX], i, j, k, n;
7     printf("Inserisci n e k (n>k): ");
8     scanf("%d %d", &n, &k);
9     printf("Inserisci %d interi: ", n);
10    for (i=0; i<n; i++) {
11        scanf("%d", &A[i]);
12        C[i] = i;
13    }
14    for (i=0; i<k; i++)
15        printf("%d ", A[C[i]]);
16    while (i >= 0) {
17        i = k-1;
18        while (i >= 0 && C[i] == n-(k-i))
19            i--;
20        if (i >= 0) {
21            C[i] = C[i] + 1;
22            for (j=i+1; j<k; j++)
23                C[j] = C[j-1]+1;
24            for (j=0; j<k; j++)
25                printf("%d ", A[C[j]]);
26        }
27    }
28    return(0);
29 }
```

Soluzione



Esercizio n. 4 (max 6 punti)

Codificare in linguaggio C un programma che implementi l'algoritmo descritto dal seguente diagramma di flusso. Nella codifica porre attenzione anche alla definizione delle variabili.



Soluzione

```
1 #include <stdlib.h>
2 #include <stdio.h>
3 #define MAX 50
4
5 int main(void) {
6     int A[MAX], i, j, n, k, p, r=0;
7     char C[MAX];
8     scanf("%d %d", &n, &k);
9     for (i=0; i<n; i++)
10         scanf("%d", &A[i]);
11     for (i=0; i<k; i++)
12         scanf("%c", &C[i]);
13     for (i=0; i<k; i++) {
14         if (C[i] == '*') {
15             p = 1;
16             for (j=0; j<n; j++)
17                 p = p * A[j];
18         } else {
19             p = 0;
20             for (j=0; j<n; j++)
21                 p = p - A[j];
22         }
23         r = r + p;
24     }
25     printf("%d\n", r);
26     return(0);
27 }
```