

Prima prova di esonero – 7 novembre 2006

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.

Letti in input tre interi positivi n , h e k , con $h < k$, generare un vettore V di n numeri interi casuali compresi tra h e k (estremi inclusi); stampare il vettore V . Costruire un altro vettore W di n elementi, i cui elementi w_i siano dati dal numero di elementi di V multipli di v_i . Stampare il vettore W .

Esempio Supponiamo che i due interi letti in input siano $n = 12$, $h = 2$ e $k = 18$. Sia inoltre $V = (10, 6, 5, 2, 7, 9, 3, 5, 15, 18, 17, 18)$ il vettore di $n = 12$ numeri interi casuali compresi tra $h = 2$ e $k = 18$. Allora il vettore W sarà il seguente: $W = (1, 3, 4, 5, 1, 3, 6, 4, 1, 2, 1, 2)$.

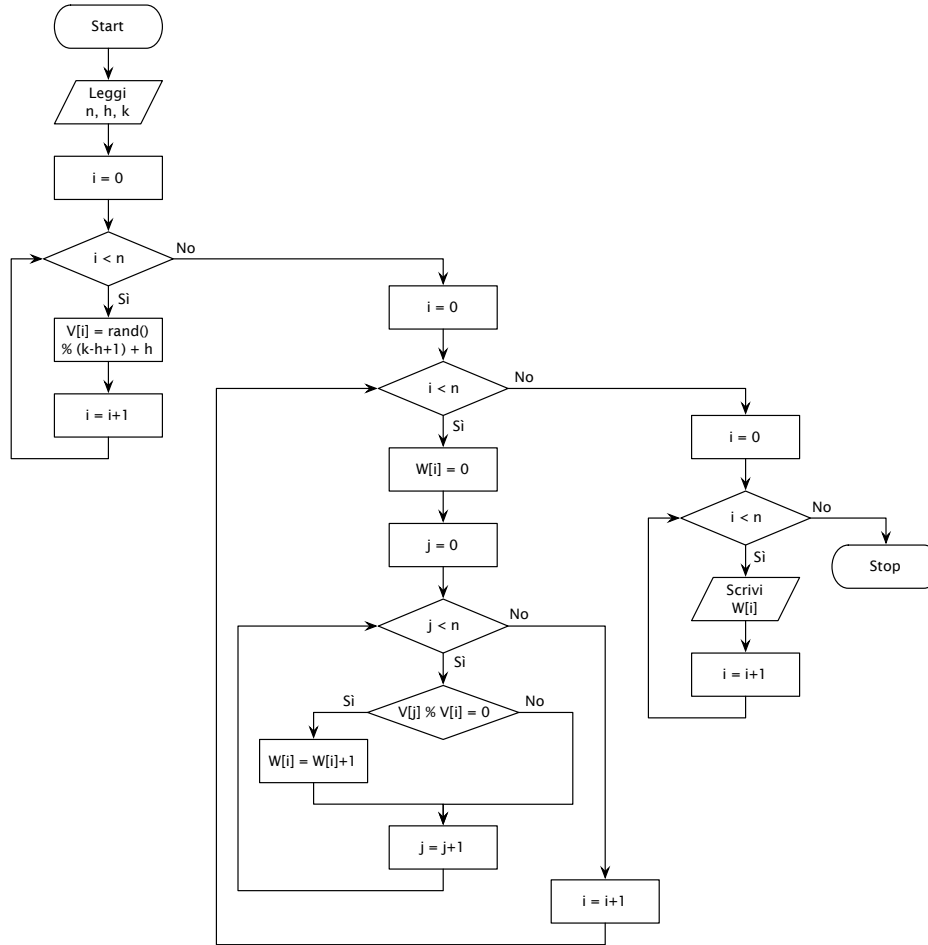
Soluzione

Pseudo-codifica dell'algoritmo

```
1: leggi  $n, h, k$ 
2: per  $i = 0, 1, 2, \dots, n - 1$  ripeti
3:    $v_i = (\text{rand}() \bmod (k - h + 1)) + h$ 
4: fine-ciclo
5: per  $i = 0, 1, 2, \dots, n - 1$  ripeti
6:    $w_i = 0$ 
7:   per  $j = 0, 1, 2, \dots, n - 1$  ripeti
8:     se  $v_j \bmod v_i = 0$  allora
9:        $w_i = w_i + 1$ 
10:    fine-condizione
11:  fine-ciclo
12: fine-ciclo
13: per  $i = 0, 1, 2, \dots, n$  ripeti
14:   scrivi  $w_i$ 
15: fine-ciclo
16: fermati
```

¹Per superare la prova di esonero e poter sostenere la successiva è necessario ottenere almeno 18 punti.

Diagramma di flusso



Codifica in linguaggio C

```
#include <stdlib.h>
#include <stdio.h>
#include <time.h>
#define MAX 100

void stampa_array(int A[], int n) {
    int i;

    for (i=0; i<n; i++)
        printf("%d ", A[i]);
    printf("\n");
    return;
}

int genera_array(int A[]) {
    int i, n, h, k;

    printf("Inserisci il numero di elementi: ");
    scanf("%d", &n);
    printf("Inserisci gli estremi dell'intervallo h e k: ");
    scanf("%d %d", &h, &k);
    srand((unsigned)time(NULL));
    for (i=0; i<n; i++)
        A[i] = (rand() % (k-h+1)) + h;
    return(n);
}

void costruisci_array(int A[], int B[], int n) {
    int i, j;

    for (i=0; i<n; i++) {
        B[i] = 0;
        for (j=0; j<n; j++) {
            if (A[j] % A[i] == 0)
                B[i]++;
        }
    }
    return;
}

int main(void) {
    int n, V[MAX], W[MAX];

    n = genera_array(V);
    stampa_array(V, n);
    costruisci_array(V, W, n);
    stampa_array(W, n);
    return(0);
}
```

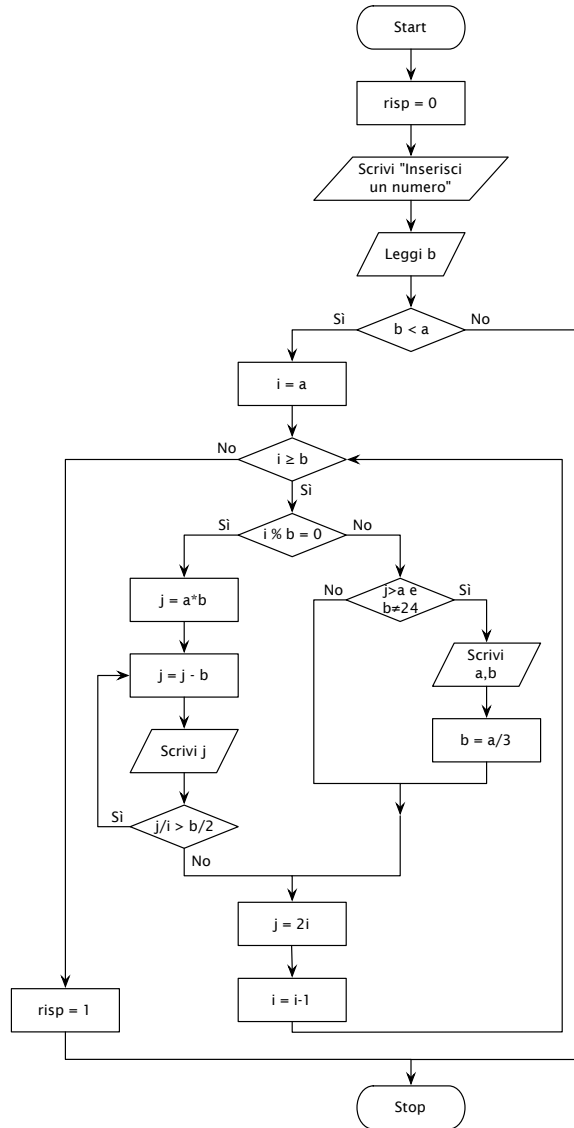
Esercizio n. 2 (max 6 punti)

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

```
void calcolaRisultato(int a) {
    int i, j, b, risp = 0;

    printf("Inserisci un numero intero minore di %d: ", a);
    scanf("%d", &b);
    if (b < a) {
        for (i=a; i>=b; i--) {
            if (i%b == 0) {
                j = a*b;
                do {
                    j = j - b;
                    printf("%5d ", j);
                } while (j/i > b/2);
            } else {
                if (j>a && b!=24) {
                    printf("%d %d\n", a, b);
                    b = a/3;
                }
            }
            j = 2*i;
        }
        risp = 1;
    }
    return(risp);
}
```

Soluzione



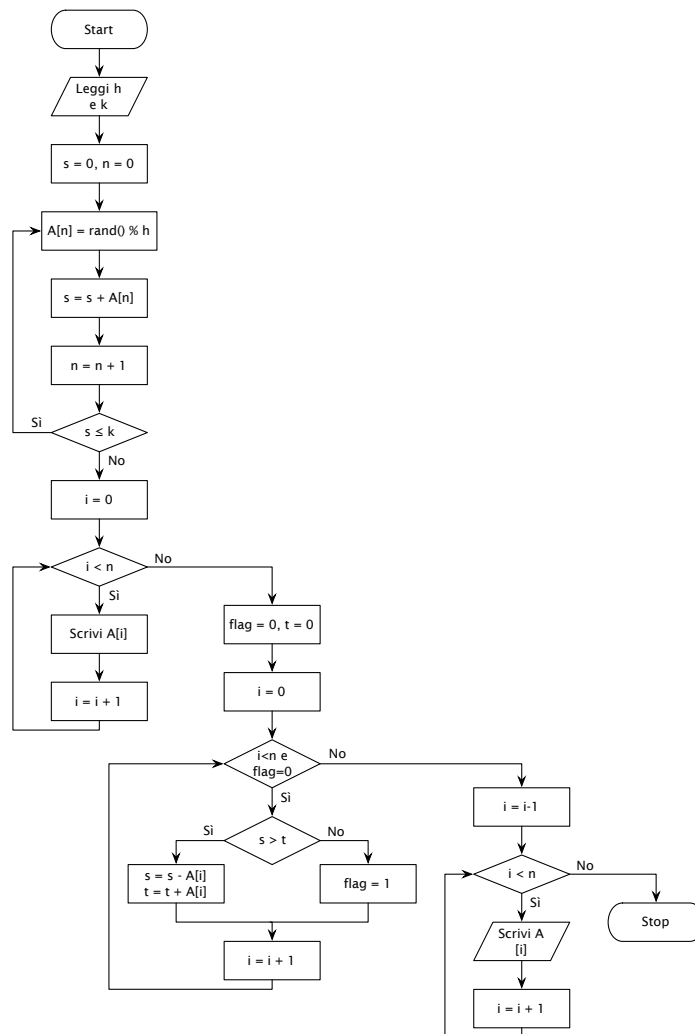
Esercizio n. 3 (max 8 punti)

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

- 1: leggi due interi $h, k > 0$, con $h < k$
- 2: costruisci un vettore A di numeri casuali interi positivi minori di h finché la loro somma non supera k
- 3: stampa A
- 4: sia n il numero di elementi di A e s la somma degli elementi a_0, \dots, a_{n-1}
- 5: $i = 0, t = 0, flag = 0$
- 6: se $s > t$ allora $s = s - a_i, t = t + a_i$ altrimenti $flag = 1$
- 7: $i = i + 1$
- 8: se $i < n$ e $flag = 0$ allora vai al passo 6
- 9: stampa gli elementi di A da quello di indice $i - 1$ fino all'ultimo
- 10: fermati

Soluzione

Diagramma di flusso



Codifica in linguaggio C

```
#include <stdlib.h>
#include <stdio.h>
#include <time.h>
#define MAX 100

void stampa_array(int A[], int n) {
    int i;

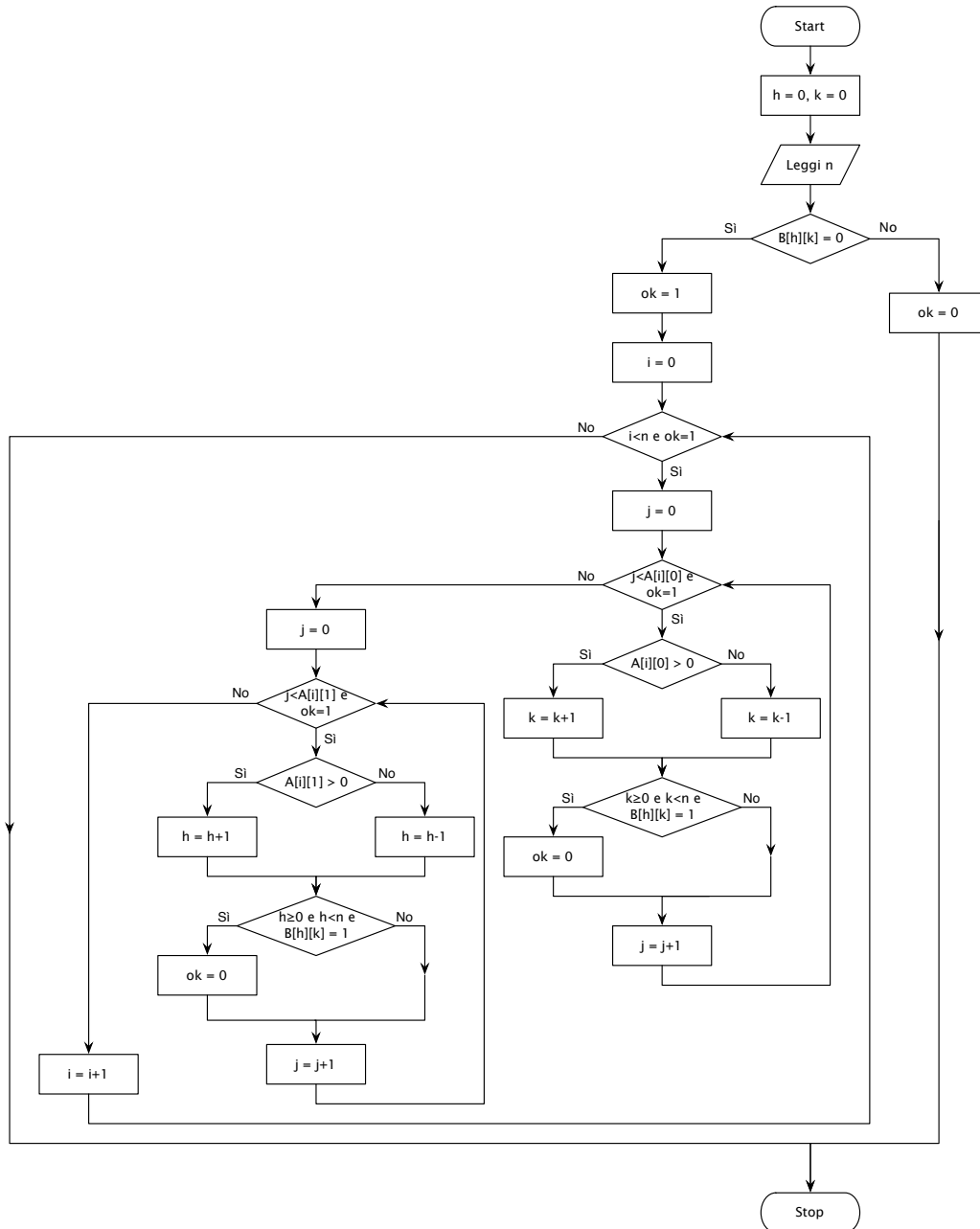
    for (i=0; i<n; i++)
        printf("%d ", A[i]);
    printf("\n");
    return;
}

int main(void) {
    int i, h, k, n=0, s=0, t=0, flag=0, A[MAX];

    printf("Inserisci due interi positivi: ");
    scanf("%d %d", &h, &k);
    srand((unsigned)time(NULL));
    do {
        A[n] = rand() % h;
        s = s + A[n];
        n++;
    } while (s <= k);
    stampa_array(A, n);
    for (i=0; i<n && flag==0; i++) {
        if (s>t) {
            s = s-A[i];
            t = t+A[i];
        } else {
            flag = 1;
        }
    }
    i = i-1;
    while (i<n) {
        printf("%d ", A[i]);
        i++;
    }
    return(0);
}
```

Esercizio n. 4 (max 6 punti)

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



Soluzione

```
int funzione(int A[MAX][2], B[MAX][MAX]) {
    int i, j, h=0, k=0, ok, n;

    scanf("%d", &n);
    if (B[h][k] == 0) {
        ok = 1;
        for (i=0; i<n && ok==1; i++) {
            for (j=0; j<A[i][0] && ok==1; j++) {
                if (A[i][0] > 0)
                    k++;
                else
                    k--;
                if (k >= 0 && k < n && B[h][k] == 1)
                    ok = 0;
            }
            for (j=0; j<A[i][1] && ok==1; j++) {
                if (A[i][1] > 0)
                    h++;
                else
                    h--;
                if (h >= 0 && h < n && B[h][k] == 1)
                    ok = 0;
            }
        }
    } else {
        ok = 0;
    }
    return(ok);
}
```