

## Prima prova di esonero – 5 novembre 2007

### 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.

Per non farsi scoprire dal professore di informatica mentre si scambiano la soluzione del compito e per non rischiare di vedersi annullato l'esame, Furio e Magda hanno progettato un astuto sistema per cifrare i messaggi: ad ogni lettera dell'alfabeto hanno associato un numero intero univoco; invece di scambiarsi il messaggio "in chiaro" gli basterà tradurlo in una sequenza di numeri ottenendo un messaggio in codice cifrato. Si scriva un programma, che letta la sequenza di numeri corrispondenti ad ogni lettera dell'alfabeto (il cifrario) e letta la sequenza cifrata, produca una stringa  $s$  con il testo in chiaro. Stampare la stringa  $s$ .

**Esempio** Supponiamo che il cifrario sia il seguente: "a"=18, "b"=5, "c"=1, "d"=38, "e"=21, "f"=17, "g"=14, "h"=128, "i"=7, "j"=24, "k"=3, "l"=2, "m"=50, "n"=30, "o"=25, "p"=11, "q"=8, "r"=12, "s"=200, "t"=150, "u"=97, "v"=80, "w"=71, "x"=69, "y"=96, "z"=42. Allora la sequenza (18, 7, 97, 150, 25) deve essere tradotta nella stringa  $s = \text{"aiuto"}$ .

### Soluzione

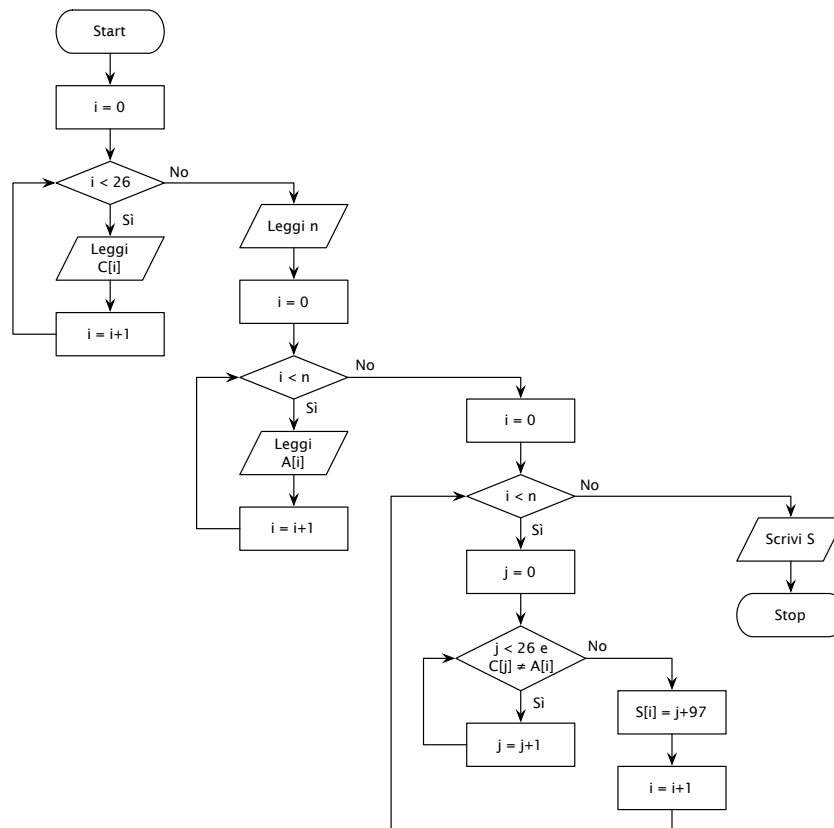
#### Pseudo-codifica dell'algoritmo

- 1: **per ogni**  $i = 0, 1, \dots, 26$  **ripeti**
- 2: leggi in input il codice numerico associato al carattere il cui codice ASCII è  $97 + i$  e memorizzalo in  $C_i$
- 3: **fine-ciclo**
- 4: leggi il numero di elementi della sequenza cifrata e memorizzalo in  $n$
- 5: **per**  $i = 0, 1, \dots, n - 1$  **ripeti**
- 6: leggi l'elemento  $A_i$  della sequenza cifrata
- 7: **fine-ciclo**
- 8: **per**  $i = 0, 1, \dots, n - 1$  **ripeti**
- 9: scorrendo il vettore  $C$  individua l'indice  $j$  tale che  $C_j = A_i$  (significa che  $A_i$  è il codice della  $j$ -esima lettera dell'alfabeto)
- 10: sia  $S_i$  l' $i$ -esimo carattere della stringa decodificata in chiaro; assegna ad  $S_i$  la lettera il cui codice ASCII è dato da  $97 + j$
- 11: **fine-ciclo**
- 12: stampa la stringa  $S$

---

<sup>1</sup>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.

## Diagramma di flusso



## Codifica in linguaggio C

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

void leggi_codice(int v[]) {
    int i;
    printf("Inserisci il codice per ciascuna lettera dell'alfabeto: ");
    for (i=0; i<26; i++) {
        printf("Codice di '%c': ", i+97);
        scanf("%d", &v[i]);
    }
    return;
}

int leggi_array(int v[]) {
    int i, n;
    printf("Numero di elementi: ");
    scanf("%d", &n);
    printf("Inserisci %d numeri interi: ", n);
    for (i=0; i<n; i++)
        scanf("%d", &v[i]);
    return(n);
}
```

```

void decodifica(int c[26], int a[], char s[], int n) {
    int i, j;
    for (i=0; i<n; i++) {
        for (j=0; j<26 && c[j] != a[i]; j++);
        s[i] = j+97;
    }
    s[i] = '\0';
    return;
}

int main(void) {
    int C[26], A[MAX], n;
    char S[MAX];
    leggi_codice(C);
    n = leggi_array(A);
    decodifica(C, A, S, n);
    printf("La stringa in chiaro e' '%s'\n", S);
    return(0);
}

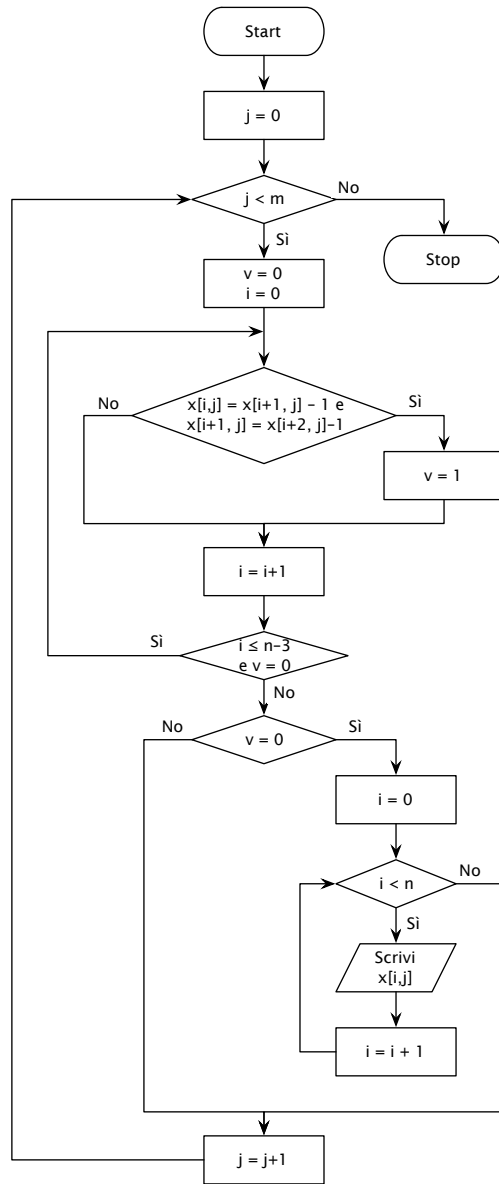
```

## 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 x[MAX][MAX], int n, int m) {
    int i, j=0, v;
    while (j<m) {
        v = 0;
        i = 0;
        do {
            if (x[i][j] == x[i+1][j]-1 && x[i+1][j] == x[i+2][j]-1)
                v = 1;
            i++;
        } while (i<=n-3 && v==0);
        if (v == 0) {
            for (i=0; i<n; i++)
                printf("%d ", x[i][j]);
            printf("\n");
        }
        j++;
    }
    return;
}
```

# 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  $n$
- 2: genera un vettore  $V$  di  $n$  interi casuali compresi tra  $-10$  e  $20$  (estremi inclusi:  $-10 \leq V_i \leq 20$  per  $i = 0, 1, \dots, n-1$ )
- 3:  $k = 0, i = 0$
- 4: se  $k > 5$  o  $i \geq n$  allora vai al passo 13, altrimenti prosegui con il passo 5
- 5:  $s = 0$
- 6:  $s = s + V_i$
- 7: se  $s < 0$  allora  $s = -s$
- 8:  $i = i + 1$
- 9: se  $s < 5$  e  $i < n$  allora vai al passo 6, altrimenti prosegui con il passo 10
- 10: scrivi  $s$
- 11:  $k = k + 1$
- 12: vai al passo 4
- 13: scrivi  $k$
- 14: fermati

### Soluzione

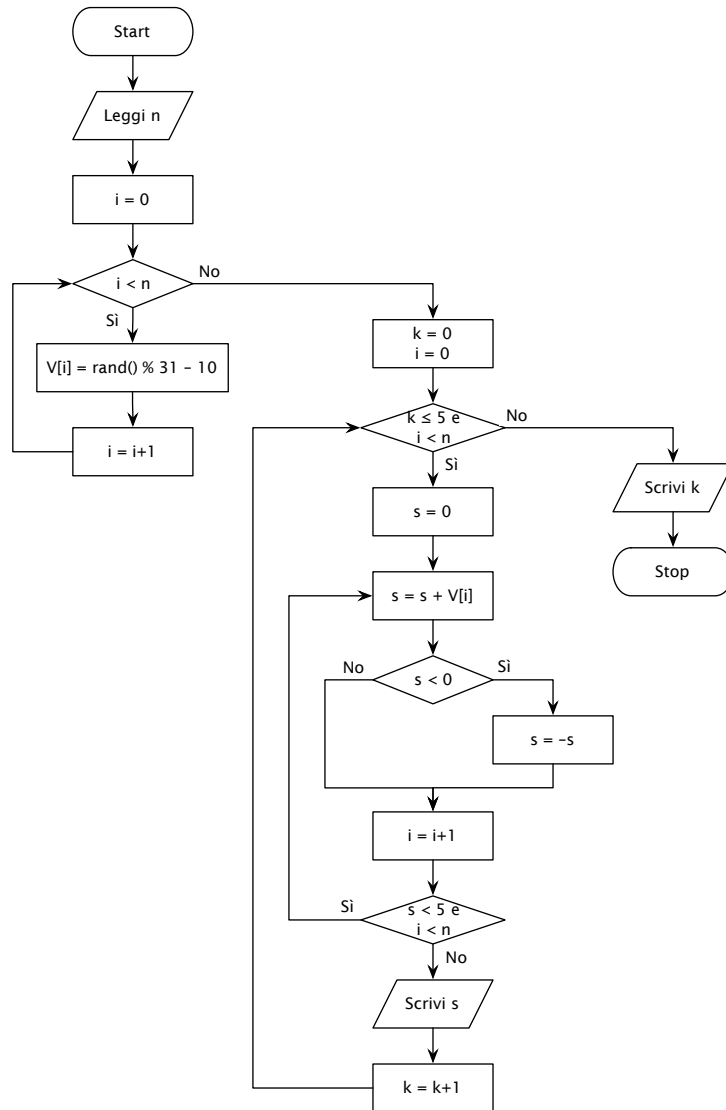
#### Codifica in linguaggio C

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

int generaVettore(int V[]) {
    int i, n;
    printf("Numero di elementi: ");
    scanf("%d", &n);
    srand((unsigned)time(NULL));
    for (i=0; i<n; i++)
        V[i] = rand() % 31 - 10;
    return(n);
}

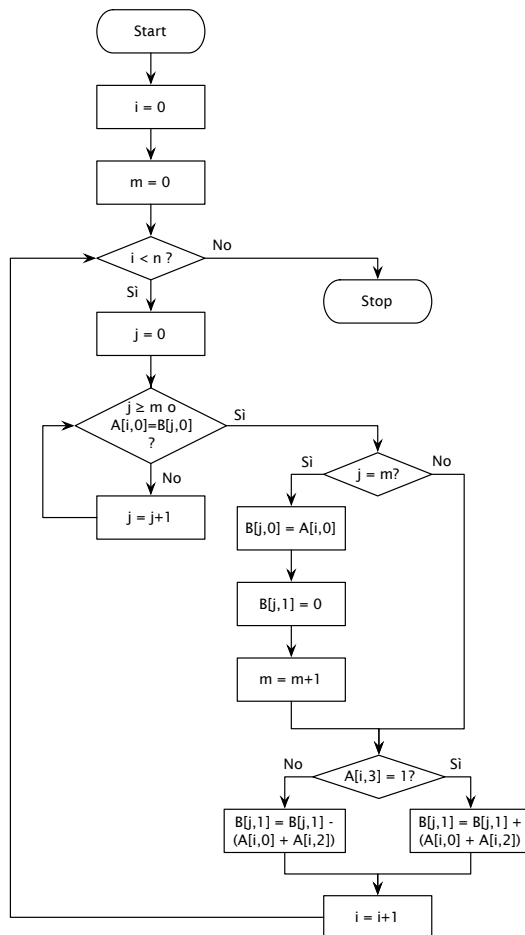
int main(void) {
    int V[100], n, k, i=0, s;
    n = generaVettore(V);
    for (k=0; k<=5 && i<n; k++) {
        s = 0;
        do {
            s = s + V[i];
            if (s<0)
                s = s * (-1);
            i++;
        } while (s<5 && i<n);
        printf("%d\n", s);
    }
    printf("%d\n", k);
    return(0);
}
```

## Diagramma di flusso



## 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

```
void funzione(int A[MAX][4], int B[MAX][2], int n) {
    int i, j, m=0;
    for (i=0; i<n; i++) {
        for (j=0; j<m && A[i][0]!=B[j][0]; j++);
        if (j == m) {
            B[j][0] = A[i][0];
            B[j][1] = 0;
            m++;
        }
        if (A[i][3] == 1)
            B[j][1] = B[j][1] + A[i][0] + A[i][2];
        else
            B[j][1] = B[j][1] - A[i][0] - A[i][2];
    }
    return;
}
```