Esercitazione 1
Conversione di base

A.A. 2018-19

Esercizio 1

(M) Siano assegnati i due M-script file conv10b.m e conv10f.m (in allegato).

1. Convertire gli M-script file a M-function file con chiamata rispettivamente:

```
function [st]=conv10b(n,b);
function [st]=conv10f(n,b,k);
```

2. Usando le function precedenti, scrivere un programma che esegua la conversione di un numero reale qualunque da base 10 a base b.

(T) Convertire il numero -2435.67 da base 10 a base 2, 8, 16.

Esercizio 2

(M) Scrivere un programma che, data in input da tastiera una stringa, la cripti usando la seguente procedura: convertire il codice ASCII di ogni carattere costituente la stringa da base 10 a base binaria usando l'algoritmo della *massima potenza contenuta*. L'algoritmo di conversione deve essere un M-function file con chiamata

```
[st]=conv10_2(n);
```

Controllare il risultato con la function MatLab dec2bin(n).
Es. INPUT: 'Hello'. OUTPUT: 1001000 - 1100101 - 1101100 - 1101100 - 1101111

(T) Convertire i numeri 278.5 e 105.3433 da base 10 a base 2, 8, 16.

Esercizio 3

(M) Due programmi devono dialogare fra di loro: l'output del primo è l'input del secondo. Il primo programma restituisce l'output in base β_1, mentre il secondo accetta input in base β_2. Dato l'intero $\beta_1 > 1$, scrivere un M-script file che

1. accetti una stringa che esprima il numero in base β_1, controllandone la correttezza;
2. converta la stringa in un numero n in base 10 (usando la funzione `polyval` per l'implementazione dello schema di Horner)
3. converta il numero n in base β_2

Es. INPUT: $\beta_1 = 2$, numero = "1000", $\beta_2 = 10$. OUTPUT: 8

(T) Convertire il numero 73564.123 da base 8 a base 7.

Esercizio 4

(M) In alcuni linguaggi di programmazione i colori sono rappresentati da una tripletta esadecimale (*hex triplet*). Ad esempio, il colore rosso nel linguaggio HTML è rappresentato da $FF0000$, mentre in coordinate RGB è $(255, 0, 0)$. La traduzione viene fatta nel seguente modo:

- FF $\rightarrow$ 255 (canale rosso);
- 00 $\rightarrow$ 0 (canale verde);
- 00 $\rightarrow$ 0 (canale blu).

Quindi le prime due cifre in base esadecimale rappresentano il livello di rosso, le due centrali il livello di verde e le due finali il livello di blu.
Utilizzare la function di MatLab `polyval` per eseguire la conversione dei codici dei colori da hex triplet a coordinate RGB. Il numero dato in input deve essere introdotto da tastiera.

(T) Convertire il numero 8E38C da base 16 a base 7.

Esercizio 5

(M) Usando la M-function `horner.m` in appendice eseguire la conversione di un numero reale scritto come stringa da base b a base 10.

(T) Convertire 10.111011 e 10101.001 da base 2 a base 10.

Esercizio 6

(M) Scrivere un M-script file che dato un polinomio di grado n, calcoli il valore del polinomio e di tutte le sue derivate fino all'ordine $n+1$ in corrispondenza di un numero reale a assegnato. Utilizzare la function `horner.m` e poi controllare i risultati ottenuti utilizzando la function MatLab

```
[q,r]=deconv(p,pp)
```

ove p contiene i coefficienti del polinomio, $pp = [1 \ -a]$ (cioè i coefficienti di $x - a$), q i coefficienti del quoziente e r quelli del resto.

(T) Determinare la complessità computazionale del precedente algoritmo.

Esercizio 7

(M) Il polinomio $m(x)$ rappresenta la massa di un razzo che viene lanciato nello spazio, mentre $v(x)$ la velocità di tale razzo mentre lascia la superficie terrestre. Il prodotto delle due quantità fisiche prende il nome di *quantità di moto*. Derivando la quantità di moto, si ottiene la forza che agisce sul razzo. Calcolare la forza agente al tempo $\overline{\alpha} = 10$.
N.B. Dati due polinomi $m(x)$ e $v(x)$, per calcolare il valore della derivata del loro prodotto in α: $(mv)'(\alpha)$. Ricordarsi della regola di derivazione del prodotto:

$$(mv)'(x) = m'(x)v(x) + m(x)v'(x)$$

Utilizzare la M-function file `horner.m` in Appendice e controllare i risultati con le function `polyval` e `polyder`.

(T) Determinare la complessità computazionale del precedente algoritmo.

Allegato

M-script file conv10b.m

```
cifre=['0','1','2','3','4','5','6','7','8','9','A','B','C','D','E','F'];
% input
n=input('Fornisci un intero positivo=');
while fix(n)~=n | n<=0
   n=input('Fornisci un intero positivo=');
end;
b=input('Fornisci una nuova base (2-16)=');
while fix(b)~=b | b<=1 | b>16
   b=input('Fornisci una nuova base (2-16)=');
end;
%
% algoritmo delle divisioni successive
%
d=n; st='';
while d~=0
   q=fix(d/b);
   r=rem(d,b);
   st= [cifre(r+1) st];
   d=q;
end;
fprintf('Conversione=%s \n',st);
```

M-script file conv10f.m

```
cifre=['0','1','2','3','4','5','6','7','8','9','A','B','C','D','E','F'];
% input
n=input('Fornisci un reale positivo minore di 1=');
while n<=0 | n>=1
   n=input('Fornisci un reale positivo minore di 1=');
end;
b=input('Fornisci una nuova base (2-16)=');
while fix(b)~=b | b<=1 | b>16
   b=input('Fornisci una nuova base (2-16)=');
end;
k=input('N. di cifre della conversione=');
while k<=0
   k=input('N. di cifre della conversione=');
end;
%
% algoritmo delle moltiplicazioni successive
%
p=n; st='0.';cont=0;
while p~=0 & cont<k
   q=p*b;
   r=fix(q);
   st=cat(2,st,cifre(r+1));
   p=q-r;cont=cont+1;
```

```
end;
fprintf('Conversione=%s \n',st);
```

M-function file horner.m

```
function [q,r]=horner(p,a);
% schema di horner per la valutazione di un
% polinomio p(x) con coefficienti nel vettore p in
% corrispondenza di x=a;
% in r resta il valore della funzione e in q
% i coefficienti del polinomio quoziente di p(x)
% diviso (x-a)
%
n=length(p)-1;
q(1)=p(1);
for i=2:n+1
   q(i)=q(i-1)*a+p(i);
end;
r=q(n+1);
q=q(1:n);
```