

1. Esercizio

- **Testo:**

Si realizzi un programma nel linguaggio C che, data una sequenza di 10 interi da standard input, facendo uso di una funzione di nome **media1**, stampi tutti i valori di segno uguale all'ultimo valore della sequenza, e calcoli la media aritmetica (reale) di tali valori.

Il programma dovrà stampare il valore calcolato dalla funzione e terminare.

Impostazione soluzione: Progetto dei dati

- Necessita` di memorizzare una sequenza di interi con segno \implies **usiamo un vettore di interi:**

```
int v[10];
```

- Progettiamo la funzione `media1`:
 - lavora su un vettore \implies **prevediamo un parametro formale di tipo vettore di interi (cioe`, puntatore a intero) e uno per la sua dimensione**
 - produce un risultato **reale:**

```
float media1(int *v, int dim);
```

Codifica:

```
#include <stdio.h>
float media1(int*v, int dim);
main()
{ int V[10], i;
  for(i=0; i<10; i++)
  { printf("\nInserire l'intero n. %d:", i+1);
    scanf("%d", &V[i]);
  }
  if (V[9]==0)
    printf("L'ultimo elemento e` nullo!\n");
  else
    printf("\nRis:  %f\n", media1(V, 10));
}
```

```
/* definizione funzione media1: */  
  
float media1(int *v, int dim)  
{ int i, ultimo=dim-1, nval=1;  
  float sum;  
  printf("Ultimo elemento: %d\n", v[ultimo]);  
  sum=v[ultimo];  
/* continua.. */
```

```
/* .. continua media1 */  
for(i=0; i<ultimo; i++)  
    {    if(v[i]*v[ultimo]>0)  
        {    nval++;  
            printf("%d\n", v[i]);  
            sum+=v[i];  
        }  
    }  
sum=sum/nval;  
return sum;  
} /* fine funzione media1 */
```

2. Esercizio

- **Testo:**

Si realizzi un programma nel linguaggio C che, dati da standard input una sequenza di 8 stringhe (eventualmente contenenti spazi bianchi), e un intero L, facendo uso di una funzione di nome **f_stringhe**, stampi tutte le stringhe della sequenza la cui lunghezza è minore di L, e calcoli il valore minimo Lmin tra le lunghezze di tali stringhe.

Il programma dovrà stampare il valore minimo Lmin calcolato dalla funzione, e terminare.

Impostazione soluzione: Progetto dei dati

- Necessita` di memorizzare una sequenza di stringhe \rightsquigarrow **usiamo un vettore di 8 stringhe:**

```
typedef char stringa[20];  
typedef stringa v_str[8];  
v_str V;
```

- Progettiamo la funzione **f_stringhe**:
 - lavora su un vettore di stringhe \rightsquigarrow **prevediamo un parametro formale di tipo v_str, uno per la sua dimensione, e un parametro per la lunghezza L data.**
 - produce un risultato **intero**:

```
int f_stringhe(v_str v, int n, int l);
```

Codifica:

```
#include <stdio.h>
#include <string.h>
typedef char stringa[20];
typedef stringa v_str[8];

/*dichiarazione della funzione: */

int f_stringhe(v_str v, int n, int l);
```

```
/* definizione del main */
main()
{   v_str V;
    int L, i;
    for(i=0; i<8; i++)
    {   printf("\nInserire stringa n. %d: ", i+1);
        gets(V[i]);
    }
    printf("\n\nValore di L: ");
    scanf("%d", &L);
    printf("\nMinimo:   %d\n", f_stringhe(V, 8, L));
}
```

```
/* definizione funzione : */
```

```
int f_stringhe(v_str v, int n, int l)  
{ int i, min, lmin, k;  
  lmin=1;  
  for(i=0; i<n; i++)  
  { k=strlen(v[i]);  
    if(k<l)  
    { printf("%s\n", v[i]);  
      if (k<lmin)  
      { lmin=k;  
        min=i;  
      }  
    }  
  }  
  return lmin;  
}
```

3. Esercizio

- **Testo:**

Si realizzi un programma nel linguaggio C che, dati due interi N e M da standard input, facendo uso di una funzione di nome **multipli**, calcoli l'insieme degli interi appartenenti all'intervallo [1, N] che sono multipli di M.

Il programma, al termine della chiamata alla funzione **multipli**, dovrà stampare:

- il numero degli elementi appartenenti all'insieme,
- tutti gli elementi dell'insieme ,

e poi terminare.

Impostazione soluzione: Progetto dei dati

- Necessita` di memorizzare una sequenza di interi (da stampare dopo l'attivazione della funzione) \Rightarrow **usiamo un vettore di interi:**

```
int v[100];
```

- **Progettiamo la funzione `multipli`:**
 - assume come dati M e N: un parametro formale per ciascuno;
 - deve assegnare valori agli elementi del vettore \Rightarrow prevediamo un parametro formale di tipo vettore di interi (cioe`, **puntatore a intero**) e uno per la sua dimensione
 - produce un risultato **intero(il numero degli elementi inseriti nel vettore v):**

```
int multipli(int m,int n,int *v,int dim);
```

Codifica:

```
#include <stdio.h>
int multipli(int m,int n,int *v,int dim);
main()
{ int V[100];
  int i, M, N, inseriti;

  printf("dammi N: ");
  scanf("%d", &N);
  printf("dammi M: ");
  scanf("%d", &M);
  inseriti=multipli(M,N,V,100);
  printf("\nL'insieme contiene %d elementi.\n",inseriti);
  for(i=0; i<inseriti; i++)
    printf("%d\n", V[i]);
}
```

```
/* definizione funzione multipli: */  
  
int multipli(int m,int n,int *v,int dim)  
{ int i, ins=0;  
  for(i=1; i<n; i++)  
    if ((i%m)==0)  
      { v[ins]=i;  
        ins++;  
      }  
  return ins;  
}
```

4. Esercizio

- **Testo:**

Si realizzi un programma nel linguaggio C che, data da standard input una sequenza di 12 stringhe, facendo uso di una funzione di nome **f_stringhe**, calcoli il sottoinsieme di stringhe date tali che:

- il primo carattere inizia con l'ultimo carattere dell'ultima stringa sequenza;

e

- la lunghezza sia minore dell'ultima stringa della sequenza

Il programma, una volta terminata la chiamata di **f_stringhe**, dovrà stampare la sequenza data e il sottoinsieme ottenuto.

Impostazione soluzione: Progetto dei dati

- Necessita` di memorizzare una sequenza di stringhe \implies **usiamo un vettore di 12 stringhe:**

```
typedef char stringa[20];  
typedef stringa v_str[12];  
v_str V;
```

- Progettiamo la funzione **f_stringhe**:

- assume un vettore di stringhe come dato, e produce un vettore di stringhe come risultato \implies **prevediamo 2 parametri formali di tipo v_str e uno per la dimensione.**
- Non sappiamo a priori da quanti elementi sara` composto il sottoinsieme ottenuto, prevediamo un risultato **intero**, per restituire tale valore.

```
int f_stringhe(v_str IN, v_str OUT int n);
```

Codifica:

```
#include <stdio.h>
#include <string.h>
typedef char stringa[20];
typedef stringa v_str[12];

int f_stringhe(v_str IN, v_str OUT, int n);

main()
{   v_str V, RIS;
    int inseriti, i;
    for(i=0; i<12; i++)
    {   printf("\nInserire la stringa n. %d: ", i+1);
        gets(V[i]);
    }
    inseriti= f_stringhe(V, RIS, 12);
    /* continua.. */
```

```
/* ..continua */  
printf("\nIl sottoinsieme contiene %d  
elementi.\n", inseriti);  
printf("\nVALORI DATI: \n");  
for(i=0; i<12; i++)  
    puts(V[i]);  
printf("\nRISULTATI: \n");  
for(i=0; i<inseriti; i++)  
    puts(RIS[i]);  
}/* fine main */
```

```
int f_stringhe(v_str IN, v_str OUT, int n)
{ char C; int i, L, ins=0;
  L=strlen(IN[n-1]); /*lunghezza dell'ultima
  stringa */
  C=IN[n-1][L-1]; /*ultimo carattere
  dell'ultima stringa*/
  for(i=0; i<n-1; i++)
    if((strlen(IN[i])<L) && (IN[i][0]==C))
    { strcpy(OUT[ins], IN[i]);
      ins++;
    }
  return ins;
}
```