

Laboratorio 30/05/03

Preparazione alla verifica

Esercizio 1

```
#include <stdio.h>
int b;
void funz (int *b, int a)
    {b = a + 1;
    return(5) }
int main()
{int a, b;
  for(a=4; a<8; a++) funz (a,b) ;
  return(0) ;}
```

correggere il programma in modo che
possa essere compilato senza errori o
warning.

Esercizio 1: soluzione

```
#include <stdio.h>
int b;
int funz (int *b, int a)
    { *b = a + 1;
      return (5) ;
    }
int main()
{int a, b;
  for(a=4; a<8; a++)    funz (&a,b) ;
  return (0) ;}
```

Esercizio 2

Data la seguente definizione di tipo:

```
typedef int vett[100];
```

scrivere una funzione C che dato un parametro *A* di tipo **vett**, restituisce attraverso il valore di ritorno od un opportuno parametro il seguente risultato: 1 se *A* contiene almeno due elementi uguali, altrimenti 0.

Esercizio 2

```
#include <stdio.h>
typedef int vett[100];
int trova (vett A, int ret)
{ int i,j;
  for(i=0; (i<99) && (ret==0); i++)
    for(j=0; j<100; j++)
      if (A[i] == A[j])
        ret=1;
  return (ret); }
```

Esercizio 2

```
main()  
{vett B;  
  int i;  
  for(i=0;i<50;i++)  
    B[i]=i;  
  for(;i<100;i++)  
    B[i]=10;  
  printf("%d \n", trova(B, &i))  
}
```

main() è solo di aiuto. trova()
deve funzionare per qualsiasi
vettore in ingresso.

Esercizio 2 soluzione

```
#include <stdio.h>
typedef int vett[100];
int trova (vett A, int *ret)
{ int i,j;
  *ret=0;
  for(i=0; (i<99) && (*ret==0); i++)
    for(j=i; (j<100) && (*ret==0); j++)
      if (A[i] == A[j])
        *ret=1;
  return(*ret); }
```

Es. 3

Data la seguente definizione di tipo:

```
typedef int vett[100];
```

scrivere una funzione C che dato un parametro `A` di tipo `vett` ed un parametro `N` di tipo `int` restituisce attraverso il valore di ritorno: 100 se `A` non contiene `N`, altrimenti il numero di elementi di `A` che precedono `N`, cioè che si trovano nelle posizioni di `A` precedenti la posizione dove si trova `N`.

Es. 3

```
#include <stdio.h>
typedef int vett[100];
int Trova (vett A, int N)
{ int i, ret = 100;
  for (i=0; i<100; i++)
    if (i == N)
      ret = i;
  return (ret);
}
```

Es. 3

```
main() /*solo per provare*/
{ vett V;
  int i;
  for(i=0;i<100; i++)
    V[i]=i;
  printf("%d\t ", Trova(V, 50));
  printf("%d\n", Trova(V, 200));
  return(0);
}
```

Es. 3: soluzione

```
#include <stdio.h>
typedef int vett[100];
int Trova (vett A, int N)
{ int i,ret = 100;
  for(i=0; i<100; i++)
    if(A[i] == N)
      ret = i;
  return (ret) ;
}
```

Esercizio 4

- Il programma che segue vuole ordinare in maniera crescente un vettore rispetto al campo a della seguente struttura:

```
typedef struct {int a,b;} ELE;
```

- correggere il codice in modo da rendere possibile ciò.
- l'algoritmo usa il metodo di selezione per ordinare.

Esercizio 4

```
#include <stdio.h>
#define DIM 3
typedef struct {int a,b;} ELE;
void carica (ELE vt[]) /*corretta*/
{int i;
  for(i=0; i<DIM; i++)
    scanf ("%d%d", &vt[i].a, &vt[i].b);
}
void stampa (ELE *vt) /*corretta*/
{int i;
  for(i=0; i<DIM; i++)
    printf ("%d\t%d\n", vt[i].a, vt[i].b);}
```

Esercizio 4

```
void ordina (ELE *vt)/*errata*/
{int i, j;
  int temp;
  for (i=0; i < DIM-1; i++)
    for (j=0; j < DIM; j++)
      if (vt[i].a > vt[j].a)
        {temp      = vt[i].a;
         vt[i].a = vt[j].a;
         vt[j].a = temp;}
}
```

Esercizio 4

```
int main()  
{ ELE v[DIM];  
  
    carica(v);  
    ordina(v);  
    stampa(v);  
  
    return(0);  
}
```

Esercizio 4: soluzione

```
void ordina (ELE *vt)
{int i, j;
  ELE temp;
  for (i=0; i < DIM-1; i++)
    for (j=i; j < DIM; j++)
      if (vt[i].a > vt[j].a)
        {temp = vt[i];
         vt[i] = vt[j];
         vt[j] = temp;}
}
```


Esercizio 5

- Data la seguente struttura dati:

```
typedef struct el {int val;  
                struct el *next;}ELE;
```

- correggere le funzioni che seguono in modo che compiano l'azione richiesta.

Esercizio 5: main()

```
int main()  
{int i;  
  ELE *caput;  
  caput=NULL;  
  for(i=0;i<5;i++)  
    caput = carica(caput);  
  stampa(caput);  
  caput=cancella(caput);  
return(0);}
```

Esercizio 5: f1

```
ELE *carica(ELE *t)
{ELE *e1, *aux, *prec;

  e1 = calloc(1, sizeof(ELE));

  printf("dammi un intero: ");
  scanf("%d", &(e1->val));
  e1->next = NULL;

  prec = aux = t;
```

inserisce in ordine gli elementi
nella lista

Esercizio 5: f1

```
if (t == NULL)
    t = e1;
else
{
    while ((aux != NULL) && \
           (aux->val <= e1->val))
    {prec = aux; aux = aux -> next;}
    e1->next = aux; prec->next = e1;}
return (t);
}
```

Esercizio 5: f1 soluzione

```
ELE *carica(ELE *t)
{ELE *e1, *aux, *prec;

  e1 = calloc(1, sizeof(ELE));

  printf("dammi un intero: ");
  scanf("%d", &(e1->val));
  e1->next = NULL;

  prec = aux = t;
```

Esercizio 5: f1 soluzione

```
if (t == NULL)
    t = e1;
else
    if (t->val > e1->val)
        {e1->next = t;
         t = e1;}
    else
        {
            while ( (aux!=NULL) &&\
                    (aux->val<= e1->val) )
                {prec = aux; aux = aux -> next;}
            e1->next = aux; prec->next = e1;}
return (t);
}
```

Esercizio 5: f2

```
void stampa (ELE *t)
{
    ELE *aux;
    aux = t;

    while (aux != NULL)
        printf ("%d\t", aux->val) ;
    aux = aux -> next;
    printf ("\n") ;
}
```

stampa gli elementi della lista

Esercizio 5: f2 soluzione

```
void stampa (ELE *t)
{
    ELE *aux;
    aux = t;

    while(aux != NULL)
        {printf ("%d\t", aux->val) ;
          aux = aux -> next;}
    printf ("\n");
}
```


Esercizio 5: f3

```
ELE *cancella(ELE *t)
{ELE *el, *aux, *prec;
 int del;
 printf("dammi elemento da canc.: ");
 scanf("%d", &del);
 aux = prec = t;
 while ((aux != NULL) && (aux->val != del))
 {   prec = aux; aux = aux -> next;}
 if (aux != NULL)
   prec->next = aux->next;

 return(t);
}
```

cancella un elemento della lista

Esercizio 5: f3 soluzione

```
ELE *cancella(ELE *t)
{ELE *el, *aux, *prec;
  int del;
  printf("dammi elemento da canc.: ");
  scanf("%d", &del);
  aux = prec = t;
  while ((aux != NULL) && (aux->val != del))
    {   prec = aux; aux = aux -> next;}
  if (aux != NULL)
    {if (t == aux) t = t -> next;
     else prec->next = aux->next;
     free(aux); }
  return(t);
}
```

Esercizio 6

- Siano dati due file di testo f e g denominati 'n1.txt' e 'n2.txt' contenenti due sequenze, ordinate in modo crescente, di numeri interi disposti uno per riga. Costruire un programma in C che scrive gli elementi di f e g sul file di testo h denominato 'risultato.txt', mantenendo h ordinato in modo crescente.

Esercizio 6

Esempio:

n1.txt	n2.txt	risultato.txt
5	3	3
15	7	5
	15	7
	24	15
		15
		24

Esercizio 6

```
#include<stdio.h>
#include <stdlib.h>
void main() {
    FILE *f, *g, *h;
    int a, b;
    if ((f=fopen("n1.txt","r")) == NULL)
        exit(-1);
    if ((g=fopen("n2.txt","r")) == NULL)
        exit(-2);
    if ((h=fopen("risultato.txt","w")) == NULL)
        exit(-3);
```

Esercizio 6

```
fscanf(f, "%d", &a); fscanf(g, "%d", &b);
while((feof(f) == 0) && (feof(g) == 0))
    if (a > b)
        { fprintf(h, "%d\n", b);
          fscanf(g, "%d", &b);
          fscanf(f, "%d", &a); }
    else
        { fprintf(h, "%d\n", a);
          fscanf(f, "%d", &a);
          fscanf(g, "%d", &b); }
fclose(f); fclose(g); fclose(h);
}
```

Esercizio 6: soluzione

```
#include<stdio.h>
#include <stdlib.h>
void main() {
    FILE *f, *g, *h;
    int a, b;
    if ((f=fopen("n1.txt", "r")) == NULL)
        exit(-1);
    if ((g=fopen("n2.txt", "r")) == NULL)
        exit(-2);
    if ((h=fopen("risultato.txt", "w")) ==NULL)
        exit(-3);
```

Esercizio 6: soluzione

```
fscanf(f, "%d", &a); fscanf(g, "%d", &b);  
while((feof(f) == 0) && (feof(g) == 0))  
    if (a > b)  
        { fprintf(h, "%d\n", b);  
          fscanf(g, "%d", &b);  
          fscanf(f, "%d", &a); /*eliminare*/ }  
    else  
        { fprintf(h, "%d\n", a);  
          fscanf(f, "%d", &a);  
          fscanf(g, "%d", &b); /*eliminare*/ }
```


Esercizio 6: soluzione

```
while (feof(f) == 0)
{ fprintf(h, "%d\n", a) ;
  fscanf(f, "%d", &a) ; }
}
while (feof(g) == 0)
{ fprintf(h, "%d\n", b) ;
  fscanf(f, "%d", &b) ; }
}
fclose(f) ; fclose(g) ; fclose(h) ;
}
```