

086180 - INFORMATICA APPLICATA

A.A. 2010-11

2° semestre

M06_E – PARADIGMI MORE

Scambio di Variabili

Dati due numeri qualsiasi X e Y si metta il minimo in X e il massimo in Y

```
int X,Y;
printf("\n\n\t dammi due numeri X ed Y= ");
scanf ("%d%d",&X,&Y);

if (X>Y) {int temp;
          temp=X; X=Y; Y=temp;
        };

printf("\n\n\t min= %d   max = %d\n\n",X,Y);
```

scambia.c

```
1 #include <stdio.h>
2 int main()
3 {
4
5     int X,Y;
6     printf("\n\n\t dammi due numeri X ed Y= ");
7     scanf ("%d%d",&X,&Y);
8
9     if (X>Y) {int temp;
10             temp=X; X=Y; Y=temp;
11             };
12
13     printf("\n\n\t minimo= %d  massimo = %d\n\n",X,Y);
14
15
16     return 0;
17 }
18
```

C:\ "C:\Programmi\C-Free Standard\temp\Untitled2.ex

```
        dammi due numeri X ed Y= 23 1
        minimo= 1  massimo = 23
Premere un tasto per continuare . . . _
```

Massimo tra più numeri positivi

```
unsigned int MAX=0;
unsigned int in =1;
while (in != 0){
printf("\n\n\t dammi un numero, finisci
(0) = ");
scanf ("%d",&in);
if (MAX<in) MAX=in;
};
printf("\n\n\t massimo = %d\n\n",MAX);
```

```
1 #include <stdio.h>
2 int main()
3 {
4
5     unsigned int MAX=0;
6     unsigned int in =1;
7     while (in != 0){
8         printf("\n\n\t dammi un numero, finisci (0) = ");
9         scanf ("%d",&in);
10        if (MAX<in) MAX=in;
11        };
12    printf("\n\n\t massimo = %d\n\n",MAX);
13
14
15    return 0;
16 }
17
```

```
dammi un numero, finisci (0) = 33
dammi un numero, finisci (0) = 6
dammi un numero, finisci (0) = 333
dammi un numero, finisci (0) = 5
dammi un numero, finisci (0) = 90
dammi un numero, finisci (0) = 0
massimo = 333
e un tasto per continuare . . . _
```

Ordinare N numeri interi

- ⦿ Si tratta di prendere il minimo dei numeri e di metterli nella prima posizione
- ⦿ Quindi prendere il minimo dei rimanenti e metterli nella seconda posizione
- ⦿ Quindi sino agli ultimi due

- ⦿ Si tratta quindi di fare più volte la determinazione del minimo scambiando di posizione i valori

- I numeri devono essere tutti contemporaneamente in memoria
- Quindi prima si leggono in un array e poi si esegue il confronto

```
int vet[N];
/*leggi gli N numeri*/
for(i=0;i<N ;i++ )
{printf("\n\t dammi un intero vet[%2d] = ",i);
scanf("%d",&vet[i]);
};
/* si trova il minimo di vet e lo si mette nella
prima posizione */
for(i=1;i<N ;i++ ) /*si confrontano i 9 successivi
col primo */
{if (vet[0]>vet[i]) {int temp;
temp=vet[0];vet[0]=vet[i];vet[i]=temp;
}
}
/*si stampa il risultato */
for(i=0;i<N ;i++ )
{printf("\n il minimo è in prima posizione =
%d\n",vet[i]);
}
```

```

1 #include <stdio.h>
2 #define N 10
3 int main()
4 {int i;
5 int vet[N];
6 /*leggi gli N numeri*/
7 for(i=0;i<N ;i++ )
8 {printf("\n\t dammi un intero vet[%2d] = ",i);
9 scanf("%d",&vet[i]);
10 };
11 /* si trova il minimo di vet e lo si mette nella prima posizione */
12 for(i=1;i<N ;i++ ) /*si confrontano i 9 successivi col primo */
13 {if (vet[0]>vet[i]) {int temp;
14                     temp=vet[0];vet[0]=vet[i];vet[i]=temp;
15                     }
16 }
17 /*si stampa il risultato */
18 for(i=0;i<N ;i++ )
19 {printf("\n il minimo è in prima posizione = %d\n",vet[i]);
20 }
21     return 0;
22 }

```

```

ero vet[ 0] = 3
ero vet[ 1] = 6
ero vet[ 2] = 99
ero vet[ 3] = 3
ero vet[ 4] = 6
ero vet[ 5] = 8
ero vet[ 6] = 1
ero vet[ 7] = 34
ero vet[ 8] = -7
ero vet[ 9] = 0
posizione = -7
posizione = 6
posizione = 99
posizione = 3
posizione = 6

```

```

il minimo p in prima posizione = 8
il minimo p in prima posizione = 3
il minimo p in prima posizione = 34
il minimo p in prima posizione = 1
il minimo p in prima posizione = 0
Premere un tasto per continuare . . .

```


Ordinamento completo

- Partendo dal programma precedente si fa effettuare l'operazione del minimo per tutti i primi 9 numeri
- il decimo sarà al suo posto

```
8 {printf("\n\t dammi un intero vet[%2d] = ",i);
9 scanf("%d",&vet[i]);
10 };
11 /* si trova il minimo di vet e lo si mette nella prima posizione */
12
13 for(i=1;i<N ;i++ ) /*si confrontano i 9 successivi col primo */
14 {if (vet[0]>vet[i]) {int temp;
15                     temp=vet[0];vet[0]=vet[i];vet[i]=temp;
16                     }
17 }
18
19
20 /*si stampa il risultato */
21 for(i=0;i<N ;i++ )
```

Parte da modificare

*Al posto dello zero (prima posizione)
si mette j che varia da 0 a N-1*

```
for(j=0;j<N-1 ;j++ )
/* per tutti i primi 9 numeri */
{
for(i=j+1;i<N ;i++ )
/*si confrontano i 9 successivi col primo */
  {if (vet[j]>vet[i])
      {int temp;
       temp=vet[j];
       vet[j]=vet[i];
       vet[i]=temp;
      }
  }
}
```

```

1 #include <stdio.h>
2 #define N 10
3 int main()
4 {int i,j; /*si aggiunge un altro indice */
5 int vet[N];
6 /*leggi gli N numeri*/
7 for(i=0;i<N ;i++ )
8 {printf("\n\t dammi un intero vet[%2d] = ",i);
9 scanf("%d",&vet[i]);
10 };
11 /* si trova il minimo di vet e lo si mette nella prima posizione */
12 for(j=0;j<N-1 ;j++ ) /* per tutti i primi 9 numeri */
13 {
14     for(i=j+1;i<N ;i++ ) /*si confrontano i 9 successivi col primo */
15         {if (vet[j]>vet[i]) {int temp;
16             temp=vet[j];vet[j]=vet[i];vet[i]=temp;
17             }
18         }
19     }
20 /*si stampa il risultato */
21 for(i=0;i<N ;i++ )
22 {printf("\n Tabella ordinata = %d\n",vet[i]);
23 }
24     return 0;
25 }
26

```

```

vet[ 0] = 5
vet[ 1] = 9
vet[ 2] = 55
vet[ 3] = -3
vet[ 4] = 0
vet[ 5] = 543
vet[ 6] = 2
vet[ 7] = 2
vet[ 8] = -98
vet[ 9] = 978
= -98
= -3
= 0
= 2
= 2
= 5

```

```

Tabella ordinata = 9
Tabella ordinata = 55
Tabella ordinata = 543
Tabella ordinata = 978
Premere un tasto per continuare . . .

```