Lesson 77 Practice 20 (Max-Min)

In this lesson we want the user to enter a number n, and that is going to be the number of values he is going to enter. The program is going to determine the maximum and the minimum of these numbers.

Keep in your mind that the user can enter positive and negative numbers.

```
#include <stdio.h>
int main() {
int n, x[100], min = 0, max = 0, i;
printf("Enter the number of numbers: ");
scanf("%d", &n);
for (i = 0; i < n; i++)
scanf("%d", &x[i]);
for (i = 0; i < n; i++)
if (x[i] > max)
max = x[i];
else if (x[i] < min)
min = x[i];
printf("Max = %d\n", max);
printf("Min = %d\n", min);
```

}

input:

Enter the number of numbers: 1

10

output:

Max = 10

Min = 0

Try the code yourself: Click Here!

Here we have one number and it has to be the max and min at the same time, but the min value equals 0.

Some people assume that all numbers entered by the user are smaller than a thousand

min = 1000

```
#include <stdio.h>

int main() {

int n, x[100], min = 1000, max = 0, i;

printf("Enter the number of numbers: ");

scanf("%d", &n);

for (i = 0; i < n; i++)

scanf("%d", &x[i]);

for (i = 0; i < n; i++)

if (x[i] > max)

max = x[i];

else if (x[i] < min)
```

```
min = x[i];
printf("Max = %d\n", max);
printf("Min = %d\n", min);
}
```

input:

Enter the number of numbers: 1

10

output:

Max = 10

Min = 10

Try the code yourself: Click Here!

The program worked well, but if the user enters a number greater than 1000, an error will show up. for example :

input:

Enter the number of numbers: 1

1200

output:

Max = 1200

Min = 1000

Another error will show up, if the user enters a negative number,

input:

Enter the number of numbers: 1

-20

output:

Max =0

Min = -20

The solution is :

```
#include <stdio.h>
int main() {
int n, x[100], min, max, i;
printf("Enter the number of numbers: ");
scanf("%d", &n);
for (i = 0; i < n; i++)
scanf("%d", &x[i]);
min=max=x[0];
//suggest that the first value is the minimum and maximum
for (i = 0; i < n; i++)
if (x[i] > max)
max = x[i];
else if (x[i] < min)
min = x[i];
printf("Max = %d\n", max);
printf("Min = %d\n", min);
```

}

input:

Enter the number of numbers: 4

200

-90

10

-100

output:

Max = 200

Min = -100

Try the code yourself: Click Here!

Another problem, the user can't enter more than 100 numbers, because we declared **x**[**100**].

So no need to use an array;

```
#include <stdio.h>

int main() {

int n,x, min, max, i;

printf("Enter the number of numbers: ");

scanf("%d", &n);
```

```
for (i = 0; i < n; i++)
scanf("%d", &x);
min=max=x;
//suggest that the first value is the minimum and maximum
for (i = 1; i < n; i++){
// i starts from 1 because the user has already entered a
//number
if (x> max)
max = x;
else if (x < min)
min = x;
scanf("%d", &x);
}
printf("Max = %d\n", max);
printf("Min = %d\n", min);
}
```

input:

Enter the number of numbers: 4

900000

-2000

5

-9

output:

Max = 900000

Min = -2000

Try the code yourself: Click Here!

This program now works 100% with whatever values the user enters.

```c
#include <stdio.h>
int main(){
int x, n = 0;
while (scanf("%d", &x) && x != -1)
//because we don't know how many number
//he will enter before entering -1
n++;
printf("%d", n);
// no need for an array
```

}

input:

3

6

5

2

1

4

-1

output:

6

Try the code yourself: Click Here!

Here we didn't add the number -1, let's try counting now :

```
#include <stdio.h>
int main(){
int x, n = 1;
while (scanf("%d", &x) && x != -1)
n++;
printf("%d", n);
}
```

input:

3

6

5

2

1

4

-1

output:

7

Try the code yourself: Click Here!

If we want to calculate the **even** numbers, it will be as follows:

```
#include <stdio.h>
int main() {
int x, n = 0;
while (scanf("%d", &x) && x != -1)
if (x % 2==0)
n++;
printf("%d", n);
}
```

input:

3

6

5

2

4

1

9

8

7

-1

output:

4

Try the code yourself: Click Here!

If we want to calculate the **odd** numbers, it will be as follows:

```
#include <stdio.h>
int main() {
int x, n = 0;
while (scanf("%d", &x) && x != -1)
if (x % 2)
n++;
printf("%d", n);
}
```

input:

3

6

5

2

4

1

9

8

7

-1

output:

5

Try the code yourself: Click Here!

If we want to know if the number is **palindrome** :

```
#include <stdio.h>
int ispal(int x) {
int y = x, z=0;
while (y) {
z = z * 10 + y % 10;
y /= 10;
}
return x == z;
```

```
//if true it will return 1
}
int main() {
int x, n = 0;
while (scanf("%d", &x) && x != -1)
if (ispal(x))
n++;
printf("%d", n);
}
```

input:

121

236

598

565

-1

output:

2

Try the code yourself: Click Here!