C Language

- C is a high level language
- General purpose and structured programming language
- C is mainly used for designing operating systems

Before writing a C program you will need a compiler:

http://www.c-compiler.com/ (for Windows)
http://www.smorgasbordet.com/pellesc/ (w7x64)
http://www.cs.virginia.edu/~lcc-win32/ (32 & 64)

Writing in C

- All keywords must be lower case letters, you can use upper case letters only when declaring constants

- C is case sensitive: Accel ≠ accel

- Keywords can't be used as functions or variable names

- C programs are divided into functions.

- Each program should have one (and only one) main function: Main()

Basic structure of a C program

```
// comments

Preprocessor statements

Main()

{
        Declarations;
        Statements;
}

User defined functions
```

Comments start with " // "

To include header files (which contain predefined functions) and to include symbolic constants

Main function starts with " { " and ends with " } "

All variables and arrays are declared and may be initialized

All statements end with " ; "

Programming in C

```
// Mi primer programa en C

#include<stdio.h>

int main(void)
{
printf("Hell World\n I hate C!\n");
system("pause");
}
```

Programming in C: standard libraries

- <assert.h>
- <complex.h>
- <ctype.h>
- <errno.h>
- <fenv.h>
- <float.h>
- <inttypes.h>
- <iso646.h>
- <limits.h>
- <locale.h>
- <math.h>
- <setjmp.h>
- <signal.h>
- <stdarg.h>
- <stdbool.h>
- <stddef.h>
- <stdint.h>
- <stdio.h>
- <stdlib.h>
- <string.h>
- <tgmath.h>
- <time.h>
- <wchar.h>
- <wctype.h>

stdio.h : standard input-output header

Some of the functions declared in the stdio.h header:

printf, vprintf	used to print to the standard output stream
fprintf, vfprintf	used to print to a file
sprintf, snprintf, vsprintf, vsnprintf	used to print to a char array (C string)
perror	writes an error message to stderr
putc	writes and returns a character to a stream and advances the file position indicator for it; equivalent to fputc, except that a macro version may evaluate the stream more than once
putchar, fputchar	has the same effects as putc(stdout)
scanf, vscanf	used to input from the standard input stream

UPRM

C format specifiers

%c	The character format specifier.
%d	The integer format specifier.
%i	The integer format specifier (same as %d).
%f	The floating-point format specifier.
%e	The scientific notation format specifier.
%E	The scientific notation format specifier.
%g	Uses %f or %e, whichever result is shorter.
%G	Uses %f or %E, whichever result is shorter.
%o	The unsigned octal format specifier.
%s	The string format specifier.
%u	The unsigned integer format specifier.
%x	The unsigned hexadecimal format specifier.
%X	The unsigned hexadecimal format specifier.
%p	Displays the corresponding argument that is a pointer.
%n	Records the number of characters written so far.
%%	Outputs a percent sign.

Programming in C: Factorial using for

```
// calcula el factorial usando for

#include <stdio.h>

int main(void)
{
        int k,fact,value;
        printf("Enter the value:");
        scanf("%d",&value);
        fact=1;
        for (k=1; k<=value; k=k+1){
                fact=fact*k;
                }
        printf("The factorial of %d is %d\n",value,fact);
        system("pause");
}
```

Factorial using a recursive function

```
// calcula el factorial usando una funcion recursiva
#include<stdio.h>
int fact(int n);            //function prototype: to check the calls

int main(void)
{
        int n,f;
        printf("\nenter the number for which you want to find the factorial: ");
        scanf("%d",&n);
        f=fact(n);
        printf("\nthe factorial of the number %d is %d ",n,f);
        system("pause");
}

int fact(int n)
{
        int k;
        if(n==0)
                return(1);
        else
                k=n*fact(n-1);
        return(k);
}
```

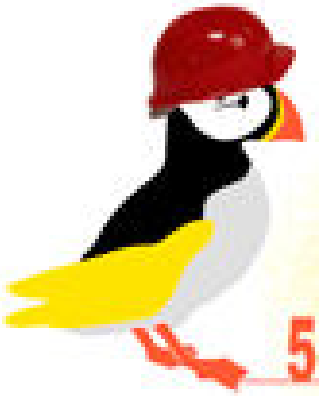
Programming in C: Factorial using while

```
// calcula el factorial usando while

#include <stdio.h>

int main(void)
{
        double fact;
        int k,value;
        printf("Enter the value:");
        scanf("%d",&value);
        fact=1;
        k=1;
        while (k<=value){
                fact=fact*k;
                k=k+1;
                }
        printf("The factorial of %d is %f\n",value,fact);
        system("pause");
}
```

Programming in C: Arrays

```
// fibonacci series 0 1 1 2 3 5 8 13 ..... using arrays

#include<stdio.h>

int main(void)
{
   int i,n,a[100];
   printf("How many terms to be display : ");
   scanf("%d",&n);
   a[0]=0;a[1]=1;

   for(i=2;i<n;i=i+1)
      a[i]=a[i-1]+a[i-2];

   printf("First %d Terms of fibonacci series \n",n);

   for(i=0;i<n;i=i+1)
       printf("%5d",a[i]);

   system("pause");
}
```

UPRM



Scilab

5.3.0 - beta 5


```
function f = fibonacci(n) // return n-th Fibonacci number

select n
    case 1 then
        f = 1
    case 2 then
        f = 1
    else
        f = fibonacci(n - 1) + fibonacci(n - 2)
end

endfunction
```