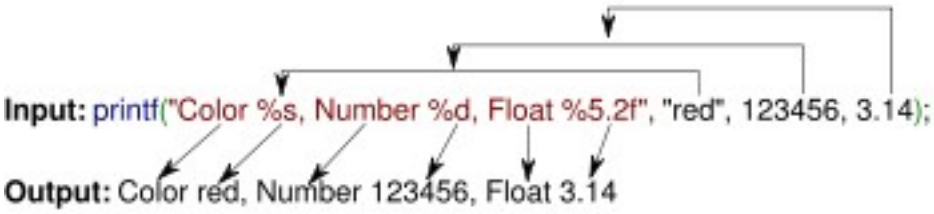

%d (or %i) int
%c char
%f float
%lf double (long float)
%s string
%e scientific notation
%x hexadecimal
%o octal

```
 /* print "the date is: 05.01.2006",
  i.e. 2- or 4-digit with leading zeros*/
   long lday   = 5;
   long lmonth = 1;
   long lyear  = 2006;
   printf("the date is: %02d.%02d.%04d\n",lday,lmonth,lyear);
```

the date is: 05.01.2006

```
/* print double value like "70.35000" */
   double dTemp = 70.35;
   printf("temperature: %5.5f\n", dTemp);
```

temperature: 70.35000

printf(“%0.5f”,(1+sqrt(5))/2)

1.618

printf(“%0.5f”,1/eps)

4.5036e+15

printf(“%15.5f”,1/eps)

4503599627370496.00000

printf(“%d”,round(pi))

3

printf(“%s”,”hello”)

hello

printf(“The array is %dx%d.”,2,3)

The array is 2x3

printf(“\n”)

Line termination character on all platforms

```
long ulID = 0x12345678;
printf("hex value: 0x%x \n", ulID);
```

hex value: 0x12345678

```
int
main()
{
        int number=5;
        char *pointer="little";

        printf("Here is a number %d and a %s word.\n", number,
pointer);
}
```

Here is a number 5 and a little word.

```
int
main()
```

```
{
        int number=5;
        char *pointer="little";

        printf("Here is a number-%4d-and a [%10s] word.\n", number,
pointer);
}
```

Here is a number-   5-and a [little] word.

```
char *psz1 = "FooBar Inc. formerly known as Foo Bar Systems Corp.";
char *psz2 = "NY";
char *psz3 = "105000";
printf("company: %-10.10s | "    /* continue on next line,*/
         "location: %10.10s | "    /* strings are joined by compiler.*/
         "turnover: %08s$\n",
         psz1,psz2,psz3
        );
```

company: FooBar Inc | location: NY | turnover: 00105000$

```
#include <stdio.h>

int main()
{
  int a = 72;
  char b = 'A';
  printf("a equals %d \n", a);
  printf("a equals %c \n", a);
  printf("b equals %d \n", b);
  printf("b equals %c \n", b);
}
```

```
a equals 72
a equals H
b equals 65
b equals A
```

```
#include <stdio.h>

int main() {
  printf("%c%c%c", 105, 111, 116);
```

```
  printf("%c%c%c", 97,  45,  115);
  printf("%c%c%c", 105, 120, 46);
  printf("%c%c%c", 99,  111, 46);
  printf("%c%c%c", 117, 107, 10);

  return 0;
}
```

```
int binarySearch(int sortedArray[], int first, int last, int key) {
   // function:
   //   Searches sortedArray[first]..sortedArray[last] for key.
   // returns: index of the matching element if it finds key,
   //         otherwise  -(index where it could be inserted)-1.
   // parameters:
   //   sortedArray in  array of sorted (ascending) values.
   //   first, last in  lower and upper subscript bounds
   //   key         in  value to search for.
   // returns:
   //   index of key, or -insertion_position -1 if key is not
   //                 in the array. This value can easily be
   //                 transformed into the position to insert it.

   while (first <= last) {
      int mid = (first + last) / 2;  // compute mid point.
      if (key > sortedArray[mid])
         first = mid + 1;  // repeat search in top half.
      else if (key < sortedArray[mid])
         last = mid - 1; // repeat search in bottom half.
      else
         return mid;     // found it. return position /////
   }
   return -(first + 1);    // failed to find key
}
```

- grapste onoma se ascii
- div