**PRACTICA N° 1:**

**1-5.-**

**#include <stdio.h>**

**int** main(**void**)

{

**float** fahr, celsius;

**int** lower, upper, step;

lower = **0**;

upper = **300**;

step = **20**;

printf("C F\n\n");

celsius = upper;

**while**(celsius >= lower)

{

fahr = (**9.0**/**5.0**) \* celsius + **32.0**;

printf("%3.0f %6.1f\n", celsius, fahr);

celsius = celsius - step;

}

**return** **0**;

}

**1-8.-**

**#include <stdio.h>**

**int** main(**void**)

{

**int** blanks, tabs, newlines;

**int** c;

**int** done = **0**;

**int** lastchar = **0**;

blanks = **0**;

tabs = **0**;

newlines = **0**;

**while**(done == **0**)

{

c = getchar();

**if**(c == **' '**)

++blanks;

**if**(c == **'\t'**)

++tabs;

**if**(c == **'\n'**)

++newlines;

**if**(c == EOF)

{

**if**(lastchar != **'\n'**)

{

++newlines;

}

done = **1**;

}

lastchar = c;

}

printf("Blanks: %d\nTabs: %d\nLines: %d\n", blanks, tabs, newlines);

**return** **0**;

}

**1-9.-**

**#include <stdio.h>**

**int** main(**void**)

{

**int** c;

**int** inspace;

inspace = **0**;

**while**((c = getchar()) != EOF)

{

**if**(c == **' '**)

{

**if**(inspace == **0**)

{

inspace = **1**;

putchar(c);

}

}

**if**(c != **' '**)

{

inspace = **0**;

putchar(c);

}

}

**return** **0**;

}

**1-10.-**

**#include <stdio.h>**

**int** main()

{

**int** c, d;

**while** ( (c=getchar()) != EOF) {

d = **0**;

**if** (c == **'\\'**) {

putchar(**'\\'**);

putchar(**'\\'**);

d = **1**;

}

**if** (c == **'\t'**) {

putchar(**'\\'**);

putchar(**'t'**);

d = **1**;

}

**if** (c == **'\b'**) {

putchar(**'\\'**);

putchar(**'b'**);

d = **1**;

}

**if** (d == **0**)

putchar(c);

}

**return** **0**;

}

**1-13.-**

**#include <stdio.h>**

**#define MAXWORDLEN 10**

**int** main(**void**)

{

**int** c;

**int** inspace = **0**;

**long** lengtharr[MAXWORDLEN + **1**];

**int** wordlen = **0**;

**int** firstletter = **1**;

**long** thisval = **0**;

**long** maxval = **0**;

**int** thisidx = **0**;

**int** done = **0**;

**for**(thisidx = **0**; thisidx <= MAXWORDLEN; thisidx++)

{

lengtharr[thisidx] = **0**;

}

**while**(done == **0**)

{

c = getchar();

**if**(c == **' '** || c == **'\t'** || c == **'\n'** || c == EOF)

{

**if**(inspace == **0**)

{

firstletter = **0**;

inspace = **1**;

**if**(wordlen <= MAXWORDLEN)

{

**if**(wordlen > **0**)

{

thisval = ++lengtharr[wordlen - **1**];

**if**(thisval > maxval)

{

maxval = thisval;

}

}

}

**else**

{

thisval = ++lengtharr[MAXWORDLEN];

**if**(thisval > maxval)

{

maxval = thisval;

}

}

}

**if**(c == EOF)

{

done = **1**;

}

}

**else**

{

**if**(inspace == **1** || firstletter == **1**)

{

wordlen = **0**;

firstletter = **0**;

inspace = **0**;

}

++wordlen;

}

}

**for**(thisval = maxval; thisval > **0**; thisval--)

{

printf("%4d | ", thisval);

**for**(thisidx = **0**; thisidx <= MAXWORDLEN; thisidx++)

{

**if**(lengtharr[thisidx] >= thisval)

{

printf("\* ");

}

**else**

{

printf(" ");

}

}

printf("\n");

}

printf(" +");

**for**(thisidx = **0**; thisidx <= MAXWORDLEN; thisidx++)

{

printf("---");

}

printf("\n ");

**for**(thisidx = **0**; thisidx < MAXWORDLEN; thisidx++)

{

printf("%2d ", thisidx + **1**);

}

printf(">%d\n", MAXWORDLEN);

**return** **0**;

}

**1-14.-**

**#include <stdio.h>**

**#define NUM\_CHARS 256**

**int** main(**void**)

{

**int** c;

**long** freqarr[NUM\_CHARS + **1**];

**long** thisval = **0**;

**long** maxval = **0**;

**int** thisidx = **0**;

**for**(thisidx = **0**; thisidx <= NUM\_CHARS; thisidx++)

{

freqarr[thisidx] = **0**;

}

**while**((c = getchar()) != EOF)

{

**if**(c < NUM\_CHARS)

{

thisval = ++freqarr[c];

**if**(thisval > maxval)

{

maxval = thisval;

}

}

**else**

{

thisval = ++freqarr[NUM\_CHARS];

**if**(thisval > maxval)

{

maxval = thisval;

}

}

}

**for**(thisval = maxval; thisval > **0**; thisval--)

{

printf("%4d |", thisval);

**for**(thisidx = **0**; thisidx <= NUM\_CHARS; thisidx++)

{

**if**(freqarr[thisidx] >= thisval)

{

printf("\*");

}

**else** **if**(freqarr[thisidx] > **0**)

{

printf(" ");

}

}

printf("\n");

}

printf(" +");

**for**(thisidx = **0**; thisidx <= NUM\_CHARS; thisidx++)

{

**if**(freqarr[thisidx] > **0**)

{

printf("-");

}

}

printf("\n ");

**for**(thisidx = **0**; thisidx < NUM\_CHARS; thisidx++)

{

**if**(freqarr[thisidx] > **0**)

{

printf("%d", thisidx / **100**);

}

}

printf("\n ");

**for**(thisidx = **0**; thisidx < NUM\_CHARS; thisidx++)

{

**if**(freqarr[thisidx] > **0**)

{

printf("%d", (thisidx - (**100** \* (thisidx / **100**))) / **10** );

}

}

printf("\n ");

**for**(thisidx = **0**; thisidx < NUM\_CHARS; thisidx++)

{

**if**(freqarr[thisidx] > **0**)

{

printf("%d", thisidx - (**10** \* (thisidx / **10**)));

}

}

**if**(freqarr[NUM\_CHARS] > **0**)

{

printf(">%d\n", NUM\_CHARS);

}

printf("\n");

**return** **0**;

}