

1. LABORATEGIA

SARRERA ETA IRTEERAKO ERAGIKETAK

1 HELBURUAK

Jarduera hau amaitzean, hurrengo egiteko gai izango zara:

- Konpiladorearen funtzionamendua ulertu.
- Aldagarriak definitu eta balio esleipena egin oinarritzko eragiketa aritmetikoekin.
- Datu sarrera eta irteera daukaten programa errazak idaztea.
- Zure lanetarako datu biltegitratze sistema bat sortzea.

2 MOTIBAZIOA

Edozein programa idazteko konpiladorea erabili behar izango dugu, horregatik komeni da bere funtzionamendua egokiro ezagutzea.

Ariketa gehien ebazpenean, erabiltzaileari datuak eskatu behar izango dizkiogu kalkuluak edo eragiketak beraiekin egiteko eta emaitzak erakusteko. Datuen sarrera eta irteera programa guztien oinarritzko atala izango da eta hau menperatzea ezinbestekoa izango da.

Ordenagailuen biltegitratze sistemak sakonki ezagutzea komeni da, gure programak berreskuratu nahi baditugu edo beste ordenagailu batetara eraman.

3 ALDAGARRIAK, SARRERAK ETA IRTEERAK C LENGOAIAN

Sarrera, irteera eta bitarteko datu guztiek aldagarri edo konstantetan gordeta egon beharko dute.

Gogoratu aldagarri bat memoriako alde bereizi bat dela, zeinari guk izen bat jarri dioguna hura erabili ahal izateko eta programan zehar bere balioa aldatzen duena. Definitzeko modua tipoa eta izena adieraziz da, puntu eta koma batez jarraituta. Adibidea:

```
int zenbakia;
```

```
float altuera;
```

Konstanteak, programan zehar beraien balioa aldatzen ez den objektuak dira eta preprozesadorearen arteztarauetan adierazten dira. Ejemplo:

```
#define PI 3,1416
```

Datuen irteerarako, printf sententzia erabiliko dugu behar diren argumentu guztiekin. Adibidea:

```
printf("\nNum-en balioa %d da eta cant-ena: %d", num, cant);
```

Datuen sarrerarako scanf sententzia erabiliko dugu behar diren argumentuekin. Adibidea:

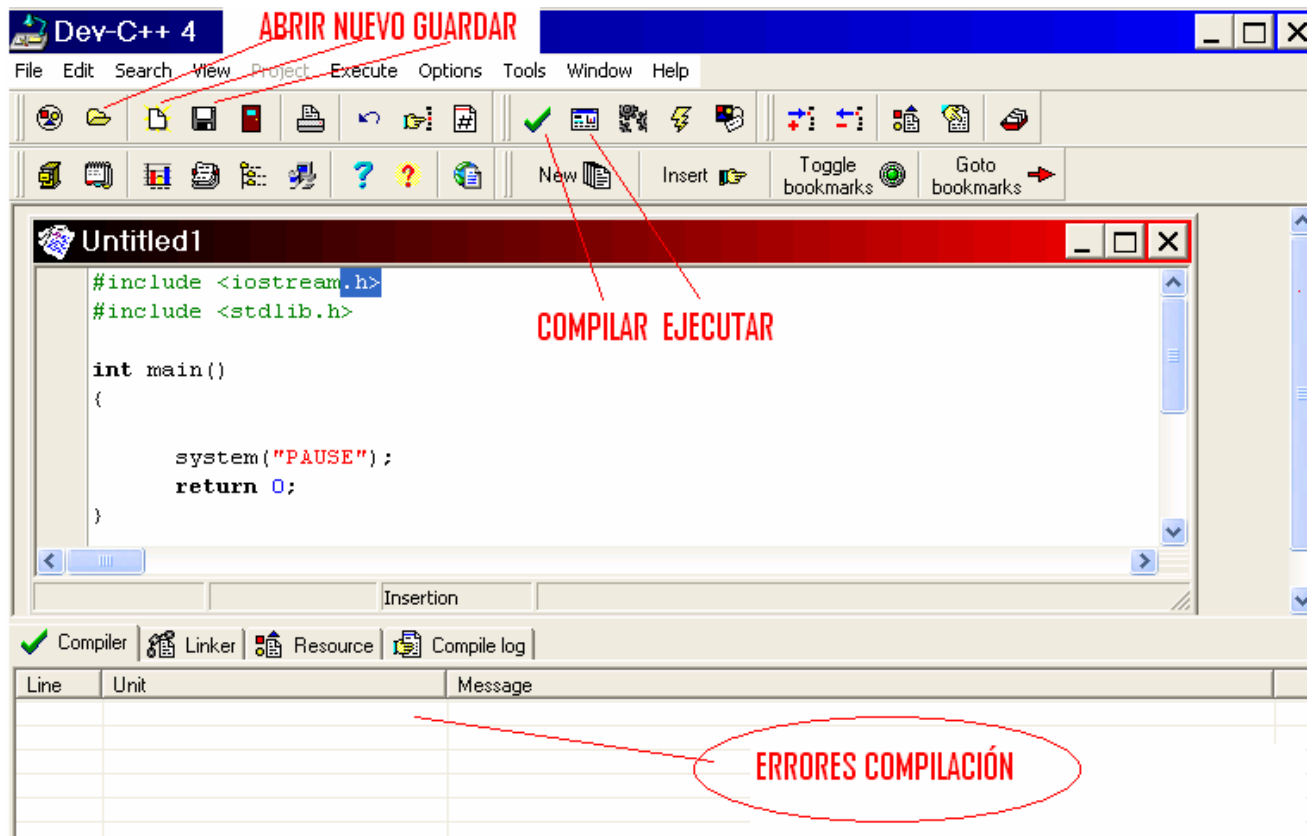
```
int a, b = 18, c;  
scanf("%d", &a);
```

4 ARIKETAK

1.1 1. ariketa

1.1.1 Helburua

Konpiladorearen oinarritzko funtzionamendua ezagutzea.



1.2 2. ariketa

1.2.1 Helburua:

Ariketa honen helburua konpiladoreari buruz ikusitakoa praktikan jartzea eta eragiketa bakoitzean aldagai bakarra erabiliz sarrera/irteera aginduak erabiltzea da.

1.2.2 1. ariketarako laguntza:

Kasu honetan eta lehenengo ariketa izateagatik, ariketaren ebazpen osoa ematen da. Beraz, konpiladorearen editorea erabiliz koptatu beharko da, akatsak ematen ez dituela ikusi, guztia gorde eta kodea egikaritu.

```
#include <stdlib.h>
#include <stdio.h>

#define PI 3.1415

void main()
{
    float erradioa,area,luzera;
    printf("Sartu erradioa: ");
    scanf("%f",&erradioa);
    luzera=2*PI*erradioa;
    printf("Luzera: %.2f",luzera);
    area=PI*erradioa*erradioa;
    printf("\nArea: %.2f\n",area);
    system("PAUSE");
}
```

1.2.3 1. enuntziatua:

Zirkulu baten erradioa irakurri eta kalkulatu eta erakutsi bere luzera eta area.

Adibidea:

```
SARTU ERRADIOA: 3,1
AREA: 30,1907
```

1.3 2 ariketa

1.3.1 Helburua:

Ariketa honen helburua gure lehen programa egitea da, aldagaien definizioa, datuen sarrera eta irteera eta konbertsioak egiteko aldagaien esleipen egokia erabiliz.

1.3.2 2. enuntziaturako laguntza:

Hurrengo adibidean gradu zentigraduetatik farenheitetara konbertsioa egiten da hurrengo formula erabiliz: $Fahrenheit = (9/5 * celsius) + 32$.

```
#include<stdio.h>
#include<stdlib.h>
/* gradu zentigraduetatik farenheitetara konbertsioa */
void main()
{
    float graduz;
    float graduf;
    printf("gradu zentigraduetatik farenheitetara konbertsioa \n");
    printf("Eman gradu zentigraduak: ");
```

```
scanf("%f",&graduz);
graduf=(9/5*graduz)+32;
printf("\nFahrenheit graduak:%f",graduf);
system("PAUSE");
}
```

1.3.3 2. enuntziatua

Eurokalkulagailu bat inplementatzen duen programa idatzi. Programak kopuru bat eurotan eskatzen du eta pezetatara bihurtzen du.

Aurreko eredua erabili euroak graduen ordezkatzeko eta pezetatan dagokion balioa aldagai bati esleitzeko, hurrengo formula jarraituz: $\text{pezetak} = \text{euroak} * 166.386$

Adibidea:

```
SARTU KOPURUA PEZETATAN: 300
300 pezetak 1,80 euro dira
```

1.4 3 ariketa

1.4.1 Helburua

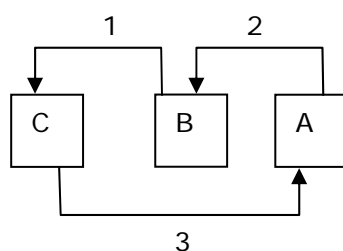
Ariketa honen helburua programa bat aldagaien definizioarekin eta datuen sarrera eta irteerarekin egitea da, eta aldagaien esleipena erabiltzea problema zehatz bat ebazteko aldagaien funtzionamendua ulertzea eta aldagai laguntzailearen kontzeptua sartzea ekartzen duena.

1.4.2 3. enuntziaturako laguntza

Aldagai baten balioa aldatzen dugunean, aurreko balioa galtzen da, hau da, nik $a=b$ esleipena egiten badut, a aldagaia b balio duena balio izatera pasatzen da eta a -ren balioa aurretik beste aldagai laguntzaile batetan gorde izan ez badut, bere hasierako balioa galduko dut.

Bi aldagaien (a eta b) balioa elkar aldatu nahi badut, orduan, pausuak hurrengoak izango dira:

1. a c aldagai laguntzailera igaro, bere balioa gordetzeko.
2. a -ren balioa txikitu b -k daukanarekin.
3. b aldatu c -k daukanarekin (hasieran a -k zeukana).



1.4.3 3. enuntziatua

Bi zenbaki teklatutik irakurri eta bere balioak elkar trukatu. OHARRA: garrantzitsua da trukatu esaten denean bere balioa ALDATU behar dela ulertzea, eta ez bakarrik mezuan aldagai bat erakustea eta bestea dela adieraztea.

Adibidea:

```
SARTU A-RENTZAT BALIOA: 5
```

```
SARTU B-RENTZAT BALIOA: 4
```

A eta B-rentzat trukaturako balioak hauek dira: A=4 y B=5

1.5 4. ariketa**1.5.1 Helburua:**

Ariketa honen helburua sarrerako zenbait datu eta baita konstante sinbolikoak erabiltzea da, kalkulu bat egiteko.

1.5.2 4. enuntziaturako laguntza

Konstante sinboliko bat izen bat moduan definitzen da, programan erabiltzen denean balio finko bat duena. Adibidez, aparteko orduengatik ordainduko dena adierazteko erabili dezakegu eta gero gure programan izena erabiliko dugu soldatak lan egindako ordu eta ordu bakoitz hauen prezio konstantearen arabera kalkulatzeko:

```
# define APARTEKOA 40
```

```
# define OHIKOA 20
```

1.5.3 4. enuntziatua

Bi datu eskatzen dituen programa bat egin: langile batek lan egindako ohiko eta aparteko orduak, eta kalkulatu zein den bere soldata osoa, hurrenez hurren orduko 20 eta 40 eurotako prezioa izanik.

OHARRA: Adibideko defineak erabili.

1.6 5. ariketa**1.6.1 Helburua:**

Ariketa honen helburua aurreko kasuetan baino kalkulu konplexuago batzuk egitea da, zenbait aldagai eta eragiketa inplikatzeko dutenak.

1.6.2 5. enuntziaturako laguntza

Hurrengo adibideak egun kopuru bat eskatzen du eta zenbat aste oso dauden adierazten du, adibidez, 15 egunetan bi aste eta egun bat adieraziko du.

```
#include <stdio.h>
#include <stdlib.h>

void main()
{
    int asteak, egunak, regunak;
    printf("Egun kopurua: ");
    scanf("%d", &egunak);
    asteak = egunak/7;
    regunak = egunak%7;
    printf("\n %d egun %d aste eta %d egun dira \n", egunak, asteak, regunak);
    system("PAUSE");
}
```

1.6.3 5. enuntziatua

Enpresa batek hiru motatako piezen produkzioaz arduratzen da P1, P2 eta P3, eta pieza bakoitzeko produkzio denborak hurrengoak dira: 3 ordu P1-entzat, 4 ordu P2-rentzat eta 5 ordu P3-rentzat.

Erabiltzaileri P1, P2 eta P3-ren kopuru bat eskatu eta kalkulatu produkzioak zenbat denbora eramango duen, hau da, zenbat aste, egun eta zein lanegun bateko ordutarako egongo da eskaria, enpresaren lan ordutegia jarraia eta 8:00-tik 16:00-arte dela jakinik.

Oharra: Erabili adibideko ideia produkzioarako ordu kopuru osotik, asteak, egunak eta gainontzeko orduak ateratzeko.