

3. LABORATEGIA

ESTRUKTURA ERREPIKAKORRAK WHILE / DO WHILE

1 HELBURUAK

Aktibitate hau amaitzean, hurrengo gaitadunak lortuko dira:

- Egitura errepikakorren erabilera eta funtzionamendua ulertu: Zer egikatzen da iterazioaren buelta bakoitzean, zein eta nola funtzionatzen den bukatzeko baldintza, baldintzari lotura dauden aldagaien eguneraketaren beharra....
- while edo do-while bat erabiltzearen arteko desberdintasunak ulertzea eta menperatzea.

2 MOTIBAZIOA

Gogoratu askotan agindu sinpleen iterazioa (errepikapena) eginez eragiketa konplexuagoak ebatzi daitezkeela. Eta hori da laborategi honetan ikusiko duguna.

while Cz

```
while (baldintza)
{
    errepikatuko den agindu multzoa
}
```

Baldintza betetzen den **bitartean**, aginduak behin eta berriz burutu. While agindua baldintza faltsua izatera pasatzen denean bukatuko da.

Do-while Cz.

```
do
{
    aginduak
}
while (baldintza);
```

Aginduak behin eta berriz burutu baldintza egiazkoa den bitartean. Do-while agindua baldintza faltsua izatera pasatzen denean bukatuko da. Baina gogoratu **BEHINTZAT BEHIN** egikarituko direla galtza artean dauden aginduak.

3 ADIBIDE ARIKETA

Ariketa honetan idatzi beharreko programak erabiltzaileari osoa eta positiboa ($1 \leq n \leq 10$) den n zenbaki bat eskatu eta lehenengo n zenbaki natural pantailan aurkeztu behar dugu. Konprobatu erabiltzaileak sartutako zenbakia egokia izatea. Ez balitz egokia izango, programa bukatuko litzateke, "Tekleatutako zenbakia ez da egokia" mezua aurkeztuz.

```

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

void main()
{
    int n, zenb;

    printf("\nOsoa eta positiboa (1<= zenbakia <=10) den zenbaki bat
tekleatu: ");
    scanf("%d", &n);

    if ((n < 1) || (n > 10))
    {
        printf("\nTekleatutako zenbakia ez da egokia.");
    }
    else
    {
        zenb = 1; /* while-eko lehenengo bueltan 1 zenbakia aztertu nahi
delako.*/
        while (zenb < n)
        {
            printf("\n%d", zenb);
            zenb = zenb + 1; /* hurrengo bueltan hurrengo zenbakia aztertzeke
*/
        } /*whilearen bukaera*/
    } /*elsearen bukaera*/
}

```

4 ARIKETAK

4.1 1. ariketa

4.1.1 Helburua

Lehenengo ariketaren helburua "indar gogorreko" ebazpenekin saiatzzea da, "indar gogorra" aukera guztiak saiatzearen zentzuan (hau da, edozein zenbaki $< n$ zatitzailea izan daiteke), iragaziz (edo, kasu honetan, imprimatuz) interesgarriak iruditzen zaizkigunak (hau da, zatitzaileak direnak).

4.1.2 Laguntza:

Zenbaki baten zatitzaileak topatzeko 1etik hasita eta erabiltzaileak sartutako zenbakira heldu arte (edo $zenb/2$ artean) dauden zenbaki guztiak sortuko ditugu modu itsu batean. Horrela sortutako zenbaki bakoitzeko egiaztatuko dugu erabiltzaileak sartutako zenbakiaren zatitzailea den ala ez (gogoratu %-ren bitartez egin dezakegula hori, $a \% b == 0$ balitz b a -ren zatitzailea dela esango genuke).

4.1.3 Lehenengo enuntziatua

Eskatu erabiltzaileari zenbaki bat sartzeko, eta kalkulatu zenbaki horren zatitzaileak.

Adibidez: pantailan agertu beharko litzatekena.

14: bere zatitzaileak 1, 2 eta 7 dira

4.2 2. ariketa

4.2.1 Helburua:

Flag edo salatarien erabilpena landu. Ebazpen desberdinak konparatu eta ondorioztatu zein den eraginkorrena zalatariarekin salataririk gabe?

4.2.2 Laguntza:

Orain (lehen ez bésala)[2..zenb-1 (edo zenb/2-1)] tartean dauden zenbaki guztiak sortuko ditugu modu itsu batean. Eta zenbaki bakoitzeko konprobatuko dugu ea erabiltzaileak sartutako zenbakiaren zatitzailea den. Horrela balitz erabiltzaileak sartutako zenbakia ez litzateke zenbaki lehena izango.

Erabili aurreko ariketan egindakoa, baina gauza batzuk aldatu beharko dituzu, adibidez orain ez gara 1 balioarekin probatzen hasiko baizik eta 2rekin....

4.2.3 2. Enuntziatua

Eskatu zenbaki bat erabiltzaileari eta erabaki ea zenbakia lehena den ala ez .

Adibidez: pantailan agertu beharko litzatekena.

13, 13 lehena da

14, 14 ez da lehena

4.3 3. ariketa

4.3.1 Helburua:

2. ariketa honen helburua **While** eta **Do-while** arteko desberdintasunak zeintzuk diren jakin eta menperatzea da.

4.3.2 Laguntza:

Gogoratu karakter bat irakurtzeko
fflush(stdin);
scanf("%c", &aldagaia);

egin behar dugula

Gogoratu ere, aldagai horren balioa *b* karakterearekin konparatu ahal izateko (aldagaia == 'b'), nun *b* komatxo artean dagoen.

Aurretik egindako ariketa erabili, zenbaki bat lehena den ala erabakitzeko.

4.3.3 3. enuntziatua

Erabiltzaileak sartutako zenbakiak lehenak diren ala ez erabakiko duen programa bat idatzi. Erabiltzaileak sartzen duen zenbaki bakoitzeko mezu hauek aterako dira:

Sartu zenbaki bat mesedez: 13

13 zenbaki lehena da

Zenbakiak sartzen jarraitu nahi duzu?(b/e)

Erabiltzaileak baietz(b) erantzungo balu programak beste zenbaki bat eskatuko lioke erabiltzaileari, honek ez duela jarraitu nahi esan arte.

Adibidez:

13 zenbakia lehena da

Zenbakiak sartzen jarraitu nahi duzu?(b/e)

b

14 zenbakia ez da lehena

Zenbakiak sartzen jarraitu nahi duzu?(b/e)

e

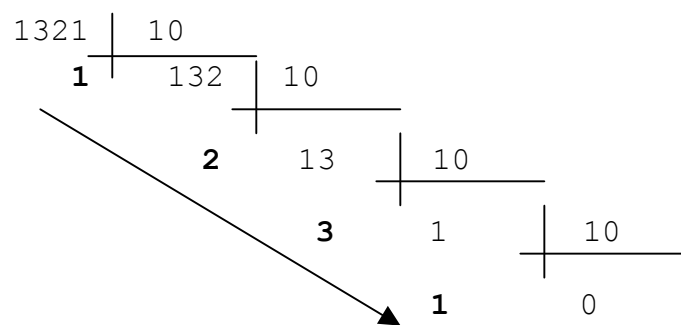
4.4 4. ariketa

4.4.1 Laguntza:

Pausuak: Zenbakiari buelta eman eta konparatu hasierako zenbakierekin.

1221ri buelta emanaz 1221 lortzen dugu, eta berdina direnez, 1221 kapikua dela esan dezakegu 1321ri buelta emanaz 1231 lortzen dugu, eta 1321 eta 1231 desberdina direnez, 1321 kapikua ez dela esan dezakegu.

Pistak: zenbaki bati buelta emateko



zatiketa bakoitzaren hondarrak hartuta eta hondar hauek akumulatuz ... 1231

4.4.2 4. enuntziatua

Eskatu erabiltzaileari zenbaki bat sartzeko, eta

konprobatu ea zenbaki hori kapikua den ala ez. Kapikua

baldin bada, inprimatu "X kapikua da". Eta ez bada, orduan inprimatu "X ez da kapikua"

Adibidez:

1221 kapikua da

1 ez da kapikua

1321 ez da kapikua

4.5 5. ariketa

4.5.1 Helburua:

Segida baten maximoa erabiltzaileak zenbakiak sartzen ari den bitartean kalkulatz jiangorela ikustea da. Horrela maximo lokalak eukiko ditugu bukaerarte.

4.5.2 Laguntza:

Aldagai auxiliar bat erabiliko dugu zenbaki berri bat sartzen den bakoitzean, momenturate maximoa izan dena gorde ahal izateko (orain_arte_max).

Adibidez:

Sartutako zenbakia 5 orain_arte_max 5
Sartutako zenbakia 7 orain_arte_max 7
Sartutako zenbakia 10 orain_arte_max 10
Sartutako zenbakia 2 orain_arte_max 10
Sartutako zenbakia -4 orain_arte_max 10
Sartutako zenbakia -15 orain_arte_max 10
Sartutako zenbakia 15 orain_arte_max 15
Sartutako zenbakia 8 orain_arte_max 15
Sartutako zenbakia 9 orain_arte_max 15
Sartutako zenbakia 0 orain_arte_max 15

4.5.3 5. enuntziatua

Eskatu erabiltzaileari zenbaki segida bat sartzeko (zenbakiz zenbaki noski). Gure programak segida horretan agertu den zenbakirik handiena inprimatuko du.

Diseinatu programa zenbakiak eskatzen jarraitzeko erabiltzaileak 0 sartu arte. Hau da 0 batekin gure programa bukatzen da.

Adibidez. Erabiltzaileak -1 3 4 -2 6 1000 2000, segida sartuz gero gure programak 2000 zenbakia inprimatu beharko luke.