

PROGRAMAZIOA

C programazio lengoaia 5

- 1) Funtzioak
- 2) Adibideak



1.1 Funtzioak

- Problemak zatitzen dituzte (programazio modularra)
- Egin behar zehatz bat izaten dute
- Parametroak jasotzen dituzte maiz
- Balio bat bueltatu dezakete (return). Emaiza asko bueltatu dezakete (erakusleak)
- Erakusleen bitartez balio ugari atzitu ditzakete (adibidez taulak) eta zenbait emaitza ere itzuli
- Liburutegietan gordeta dauden funtzioei liburutegi-funtzioak deitzen zaie

SORTU	DEITU
mota izena (parametro formalak) {aginduak;} int Berbidura (int X) {return X*X;}	Izena (parametro errealak); Z= Berbidura (4);

2007-2008

Informatikaren Oinarriak - Iker Azpeitia

2

1.2 Funtzioak Parametro errealak

- Funtzioa deitua denean pasatzen zaizkion balio zehatzak:
 - Konstanteak
 - Aldagaien balioak

```
void main ()  
{...  
Z= Berbidura (4);  
K= Berbidura (Z);  
...}
```

2007-2008

Informatikaren Oinarriak - Iker Azpeitia

3

1.3 Funtzioak Parametro formalak

- Funtzioa exekutatzen hastean memoria gunea bat hartzen dute (erazagututa daude). Funtzioaren exekuzioa bukatzean desagertzen dira
- Soilik funtzioaren gorputzean erabiltzen diren aldagaiak dira (aldagai lokalak). Beste funtzioetan daudenen desberdinak dira nahiz izen berdina izan.
- Parametro errealak jaso eta gordetzen dituzte
 - Balio bidezko parametroak
 - Erreferentzia bidezko parametroak

2007-2008

Informatikaren Oinarriak - Iker Azpeitia

4

1.4 Funtzioak Balio bidezko parametroa



- Balioa pasatzen da
- Deian erabili diren aldagaiak ez dira aldatzen

```
void main ()
{...
Z = 4;
K= Berbidura (Z);      Z == 4
...}                    K == 16
```

```
int Berbidura (int X)
{X = X* X;
return X;}
```

1.5 Funtzioak Erreferentzi bidezko parametroa



- Aldagai baten helbidea pasatzen da
- Azpiprograman egindakoak funtzio nagusiko aldagaian eragina izan dezake

```
void main ()
{...
Z=4;
Berbidetu (&Z);      Z == 16
...}
```

```
void Berbidetu (int *erakX)
{*erakX = *erakX * *erakX;}
```

1.6 Funtzioak Funtzio motak



	Parametrorik ez	Soilik balio bidezko parametroak	Erreferentzi bidezko parametroren bat
Void bueltatu	Eraitzik ez Mota1	Eraitzik ez Mota2	Eraitza asko bueltatzeko aukera (parametroak alda ditzake) Mota3
Balio bat bueltatu	Eraitza bat Mota4	Eraitza bat Mota5	Eraitza asko bueltatzeko aukera (parametroak alda ditzake) Mota6

Funtzioek beti jaso ditzakete balioak erabiltzaileengandik eta mezuak (balioak) pantailaratu.

2.1 Adibideak Funtzio mota1



Programa geldiarazteko jartzen ohi dugun kodea biltzen duen funtzioa:

```
void Geldi ()
{
puts ("\n\tSakatu tekla bat...");
getch();
}
```

2.2 Adibideak Funtzio mota2



Karaktere bat zenbait bider pantailaratzen duen funtzioa:

```
void KaraktereaPantailaratu (int a, char b)
{
for ( ; a > 0 ; a--)
    {printf ("%c", b);}
}
```

2007-2008

Informatikaren Oinarriak - Iker Azpeitia

9

2.3 Adibideak Funtzio mota3



Zenbaki baten Bateko digitua, Hamarreko digitua, Ehuneko digitua, Milako digitua eta Hamar-milako digitua bueltatzen duen funtzioa:

```
void Deskonposatu (long int Z, int *erakB, int *erakH, int *erakE, int *erakM, int *erakHM)
{
    *erakB = Z % 10;
    Z = Z/10;
    *erakH = Z % 10;
    Z = Z/10;
    *erakE = Z % 10;
    Z = Z/10;
    *erakM = Z % 10;
    Z = Z/10;
    *erakHM = Z % 10;
}
```

2007-2008

Informatikaren Oinarriak - Iker Azpeitia

10

2.4 Adibideak Funtzio mota4



Ausazko maiuskula bat bueltatzen duen funtzioa (ikus rand() liburutegi-funtzioa):

```
char MaiuskulaBat ()
{
    return ('A'+rand()%('Z'-'A'));
}
```

2007-2008

Informatikaren Oinarriak - Iker Azpeitia

11

2.5 Adibideak Funtzio mota5



Hiru zenbakien arteko handiena bueltatzen duen funtzioa:

```
float Handiena (float A, float B, float C)
{ float Max;
  if (A > B)
    { Max = A; }
  else
    { Max = B; }
  if (C > Max)
    { Max = C; }
  return Max;
}
```

2007-2008

Informatikaren Oinarriak - Iker Azpeitia

12

2.6 Adibideak Funtzio mota6



Taula bat biderkatu eta honen batura
bueltatzen duen funtzioa:

```
int BiderBatura (int * erakT, int Topea, int Bider)
{ int i, Em;
  for (i=0, Em = 0; i < Topea ; i++, erakT++)
    {*erakT = *erakT * Bider;
     Em += *erakT;}
  return Em;
}
```

2.7 Adibideak Funtzio mota6. HASIERAN



```
void main ()
{int T[3]={0,1,2}, Z=0;
 Z = BiderBatura (&T[0], 3, 4);
}

int BiderBatura (int * erakT, int Topea, int Bider)
{ int i, Em;
  for (i=0, Em = 0; i < Topea ; i++, erakT++)
    {*erakT = *erakT * Bider;
     Em += *erakT;}
  return Em;
}
```

...
65000	0	T[0]
65002	1	T[1]
65004	2	T[2]
65006	0	Z
65008		
65010		
65012		
65014		
65016		
65018		
65020		
65022		
...		

2.8 Adibideak Funtzio mota6. EXEKUZIOAN



```
void main ()
{int T[3]={0,1,2}, Z=0;
 Z = BiderBatura (&T[0], 3, 4);
}
```

```
int BiderBatura (int * erakT, int Topea, int Bider)
{ int i, Em;
  for (i=0, Em = 0; i < Topea ; i++, erakT++)
    {*erakT = *erakT * Bider;
     Em += *erakT;}
  return Em;
}
```

2.9 Adibideak Funtzio mota6. BUKAERAN



```
void main ()
{int T[3]={0,1,2}, Z=0;
 Z = BiderBatura (&T[0], 3, 4);
}
```

```
int BiderBatura (int * erakT, int Topea, int Bider)
{ int i, Em;
  for (i=0, Em = 0; i < Topea ; i++, erakT++)
    {*erakT = *erakT * Bider;
     Em += *erakT;}
  return Em;
}
```

...
65000	0	T[0]
65002	4	T[1]
65004	8	T[2]
65006	12	Z
65008		
65010		
65012		
65014		
65016		
65018		
65020		
65022		
...		