



ALT SEVİYE PROGRAMLAMA

- ÖDEV 2 -

GRUP 1

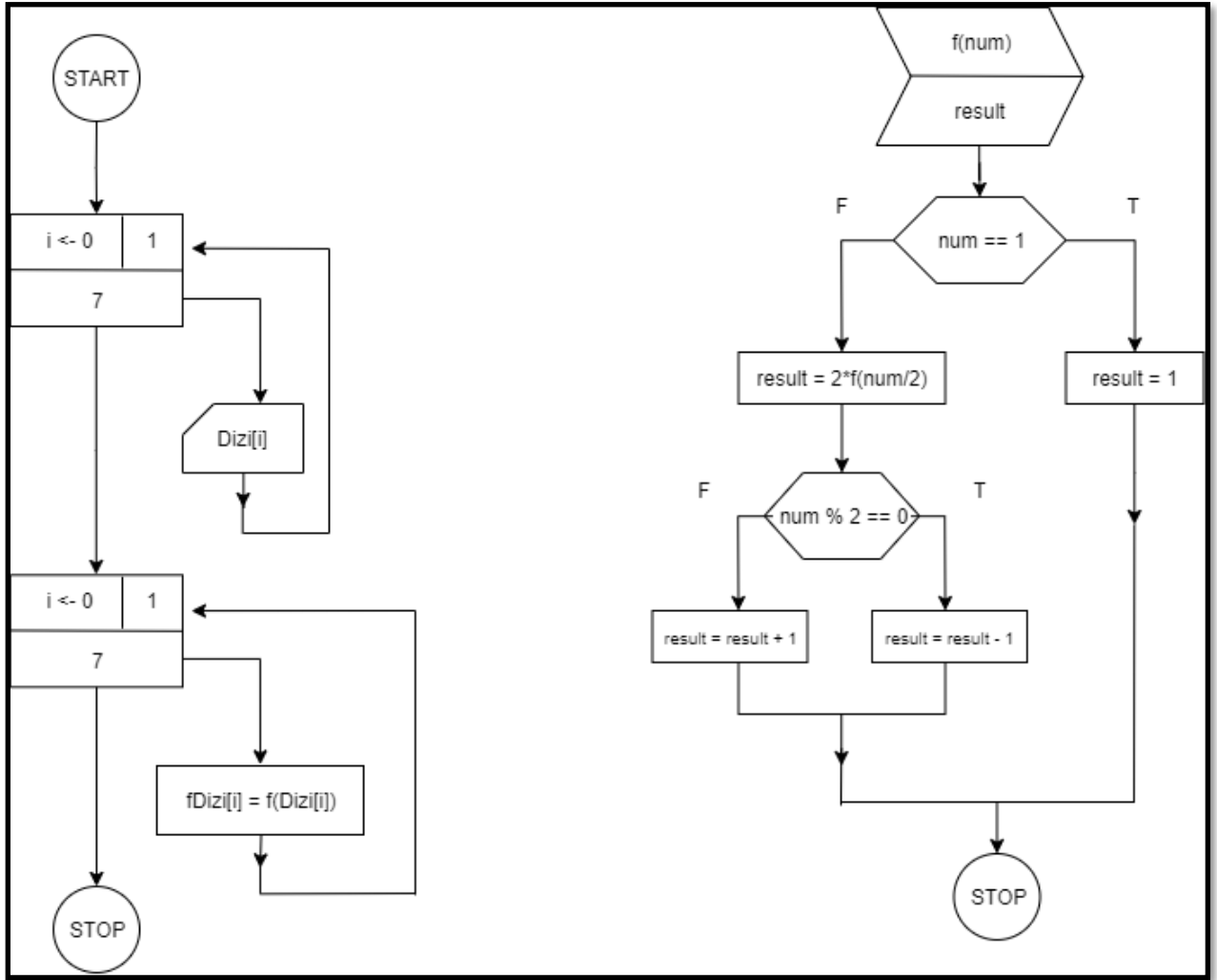
İrem ATILGAN

17061036

İrem Atılgan

05.12.2019

A)



Şekil1: Ana prosedürün ve alt prosedürün akış diyagramı

```

1  #include <stdio.h>
2
3  int f(int);
4
5  int main()
6  {
7      int dizi[] = {11,19,47,100,285,458,11111};
8      int fdizi[7] = {0};
9      int i;
10     for(i = 0; i < 7; i++)
11     {
12         fdizi[i] = f(dizi[i]);
13     }
14
15     for(i = 0; i < 7; i++)
16     {
17         printf("%d\t", fdizi[i]);
18     }
19
20     return 0;
21 }

```

```

23
24 int f(int sayi)
25 {
26     int sonuc;
27     if(sayi == 1)
28     {
29         sonuc = 1;
30     }
31     else
32     {
33         sonuc = 2*f(sayi/2);
34         if(sayi%2 != 0) sonuc++;
35         else sonuc--;
36     }
37
38     return sonuc;
39 }

```

Akış diyagramının C kodu

C)

Yığının en dolu olduğu andaki adresi (SP) = 0700:FFD8 -> 0001H (Adresin sakladığı değer)

Recursive fonksiyon çalıştıkça, yani verilen AX 1 olmadığı sürece BP ve AX değerleri stack'e sürekli PUSH edilir. Bu sırada AX yazmacı, 1 olana dek 2'ye bölünür. Bir önceki AX ve BP değerleri ise geri dönebilmek adına stack'te tutulur. AX 1 olduğunda stack'teki değerler teker teker POP edilir. Bu yüzden stack'in en dolu olduğu an ilk POP işleminden önceki andır.

	Adres	Değer
	0700:FFFE	0000H
[075A:0103]	0700:FFFC	000BH
IP+3	0700:FFFA	0160H
BP	0700:FFF8	0000H
AX	0700:FFF6	0000H
AX	0700:FFF4	000BH
AX	0700:FFF2	0005H
IP+3	0700:FFF0	0136H
BP	0700:FFDE	FFF6H
AX	0700:FFDC	0005H
AX	0700:FFEA	0005H
AX	0700:FFE8	0002H
IP+3	0700:FFE6	0136H
BP	0700:FFE4	FFEC
AX	0700:FFE2	0002H
AX	0700:FFE0	0002H
AX	0700:FFDE	0001H
IP+3	0700:FFDC	0136H
BP	0700:FFDA	FFE2H
AX	0700:FFD8	0001H

B)

075A:0100 EB 51 90 0B 00 13 00 2F - 00 64 00 1D 01 CA 01 67

075A:0110 2B 07 00 07 00 07 00 1F - 00 49 00 3B 00 95 01 CF

075A:0120 16