# HIZLI MODÜLER ÜS ALMA ALGORİTMASI İLE VERİLEN MODULER İŞLEMİ YAPAN C KODU VE EKRAN ÇIKTILARI

1)İlk olarak binary sayımızı bir diziye atma ihtiyacımız olduğu için ve binary sayımızın kaç basamaklı olduğu bilgisine ihtiyacımız olduğu için 2 adet global değişken tanımlandı.

int i = 0;

int binary\_dizi[100];

2)Kullanıcının gireceği üs değerini binary sayıya çevirecek çevirici fonksiyon olan convert\_to\_binary fonksiyonu tanımlandı. Ayrıca bu fonksiyonda binary sayının kaç basamaklı olduğunu tutacak olan i global değişkeni basamak sayısı kadar arttırılır.

int convert\_to\_binary(int dec)

{

if (dec == 0)

{

return 0;

}

else

{

i++;

return (dec % 2 + 10 \* convert\_to\_binary(dec / 2));

}

}

3)convert\_to\_binary fonksiyonu integer bir değer döndürüyordu ama bizim ihtiyacımız olan binary sayının tek tek basamak değerleriydi. Bu sebeple integer sayıyı basamak basamak diziye kopyalayan convert\_to\_array fonksiyonu tanımlandı.

void convert\_to\_array(int binary) {

int kalan;

int sayac = i-1;

for (int k = 0; k < i; k++) {

if (binary > 1) {

kalan = binary % 10;

binary\_dizi[sayac] = kalan;

sayac--;

binary = binary / 10;

}

}

if (binary == 1) {

binary\_dizi[sayac] = 1;

}

}

4)Oluşturduğumuz dizinin doğru binary sayıyı tutup tutmadığını kontrol etmek amacıyla diziyi\_yazdir fonksiyonu tanımlandı. (Sadece test amaçlı)

/\*void diziyi\_yazdir(int dizi[]) {

int k = 0;

for ( k; k< i; k++) {

printf("%d \n", dizi[k]);

}

}\*/

5)Derste verilen PseudeCode’a göre Hizli\_moduler\_us\_alma fonksiyonu tanımlandı.

a->girilen sayı değeri

mod->girilen mod değeri

Binary sayı dizisi zaten global değişkende tutulduğundan fonksiyona parametre olarak gönderme gerekliliği kalmadı.

int Hizli\_moduler\_us\_alma(int a, int mod) {

int c = 0;

int d = 1;

for (int k=0; k <i; k++) {

c = 2 \* c;

d = (d\*d) % mod;

if (binary\_dizi[k] == 1) {

c++;

d = (d\*a) % mod;

}

printf("%d. ADIM:b[%d]=%d c=%d d=%d \n",k+1,i-k-1 ,binary\_dizi[k] ,c, d);

}

return d;

}

6)Main fonksiyonda kullanıcıdan değerler istendi. Alınan üs değeri binary’e çevrildi. Binary değerin basamakları tek tek dizinin elemanlarına atandı. Daha sonra hızlı modüler üs algoritması girilen değerlere uygulandı.

int main()

{

int cikis\_kontrol;

int a, b, n;

printf("a^b(mod n) formatinda giris yapilmasi gerekmektedir. Lutfen istenilen degeri girin:\n");

printf("a degerini giriniz:");

scanf\_s("%d", &a);

printf("b degerini giriniz:");

scanf\_s("%d", &b);

printf("n degerini giriniz:");

scanf\_s("%d", &n);

int binary=convert\_to\_binary(b);

convert\_to\_array(binary);

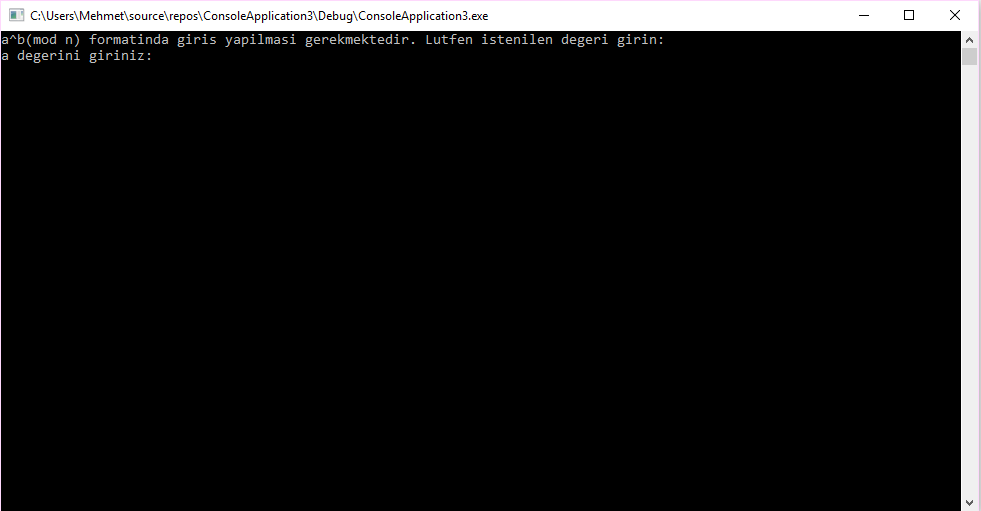
//diziyi\_yazdir(binary\_dizi);

printf("\nSonuc=%d\n",Hizli\_moduler\_us\_alma(a, n));

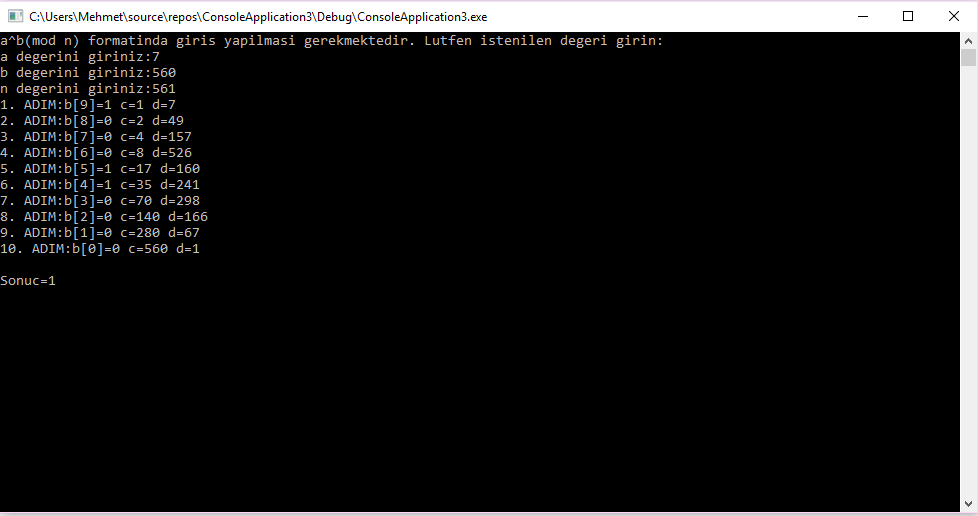
scanf\_s("%d", &cikis\_kontrol);

return 0;

}



Uygulama başladığında kullanıcıdan değerleri a^b(mod n) formatında ister. Kullanıcı sırasıyla a,b ve n değerlerini girer.



Girilen değerler (7,560,561) için önce convert\_to\_binary fonksiyonu çalışır ve b değerini (560) binary’e (1000110000) çevirir. Daha sonra bu binary değer için convert\_to\_array fonksiyonu çalışır ve binary değeri basamak basamak global değişken binary\_dizi değişkenine yazılır. Daha sonra hızlı modüler üs alma algoritması çalışır ve c, d, b[k] değerleri adım adım çıktı olarak kullanıcıya verilir. b[0] adımında hesaplanan d değeri artık bizim sonucumuzdur.