



NUMARA:	AD SOYAD:	İMZA:
---------	-----------	-------

1. Aşağıda verilen program çalıştırıldığında hangi ekran çıktısını üretir? (20p)

```
#include <stdio.h>

int main( ) {
    int x = -4;  int y = 2;  int z = 8;  int t = 5;
    int num;

    num = y+(x++)+y*z/t-x;
    num > y*(y++)+(--x)?printf("The value of num is %d",num):printf("The
value of y is %d",y);

    return 0;
}
```

The value of num is 4

2. Aşağıda verilen prosedür myFunction (2) şeklinde çağrıldığında hangi ekran çıktısını üretir? (20p)

```
procedure myFunction ( N )
if ( N > 0 )
    then ( N değerini ekrana yazdır ve myFunction'ı N-2 argümanı ile çağır)
    else ( N değerini ekrana yazdır ve
        if ( N > -1 )
            then ( myFunction'ı N+1 argümanı ile çağır ) )
```

201-1

3. Aşağıda verilen program klavyeden girilen sayıda satır içeren piramit yıldız ekran çıktısını vermektedir. Bu işlevi gerçekleştiren main() fonksiyonundaki boşlukları uygun şekilde doldurunuz. (25p)

<pre>#include <stdio.h> int main(){ int rows; scanf("%d",&rows); for (int i=1; i<=rows; i++){ for(int j=rows; j>i; j--){ printf(" "); } for (int k=1; k<=(i*2)-1; k++){ printf("*"); } printf("\n"); } return 0; }</pre>	<pre>rows=3 * *** ***** rows=5 * *** ***** ***** *****</pre>
--	---



4. n basamaklı bir sayıdaki rakamların n . kuvvetleri toplamı bu sayının kendisine eşitse bu sayılara *Armstrong sayıları* denir. (örn: $153 = 1^3 + 5^3 + 3^3$ ve $1634 = 1^4 + 6^4 + 3^4 + 4^4$ sırasıyla 3 ve 4 basamaklı birer Armstrong sayıdır). Aşağıda verilen program klavyeden okunan iki sayı ($num1$, $num2$) arasındaki Armstrong sayıları ekrana yazdırmaktadır. Programın içerdiği mantık hatalarını kod üzerinde altını çizerek tespit ediniz ve tespit ettiğiniz satır ya da satırların düzeltilmiş halini ilgili satırın yanında belirtiniz. (Not: *i*) Örnek ekran çıktısı aşağıda verilmektedir, *ii*) `pow()` fonksiyonu üst alma işlemi için kullanılmaktadır) (25p)

1 1000

Armstrong numbers between 1 and 1000 are: 2 3 4 5 6 7 8 9 153 370 371 407

```
#include <stdio.h>
#include <math.h>
int main(){
    int num1, num2, var1, var2, remainder;
    int n = 0, result = 0;

    scanf("%d %d", &num1, &num2);
    printf("Armstrong numbers between %d and %d are: ", num1, num2);

    for(int i = num1 + 1; i < num2; ++i){
        var1 = i;
        var2 = i;
        while (var1 != 0){
            var1 /= 10;
            ++n;
        }
        while (var2 != 0){
            remainder = var2 % 10;
            result = var2 + pow(remainder, i); (result=result+pow(remainder, n))
            var2 /= 10;
        }
        if (result == i){
            printf("%d ", result);
        }
        n = 0;
        result = 0;
    }
    return 0;
}
```

5. Aşağıda verilen `do-while` döngüsünü `for` döngüsüne dönüştürünüz. (10p)

```
int num=0;
do
{
    printf("%d ", num);
    num=num+2;
}
while (num<=100);
```

```
int num=0;
for (num=0; num<=100; num=num+2)
    printf("%d ", num);
```