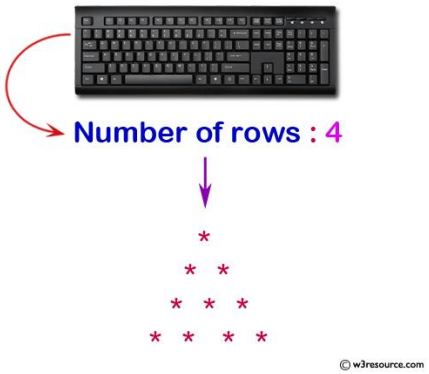
ALGORİTMALAR VE PROGRAMLAMA II – HAFTA 2

ÖRNEKLER

Örnek 1: Yıldız işaretli bir piramit gibi bir desen oluşturmak için bir C programı yazın.

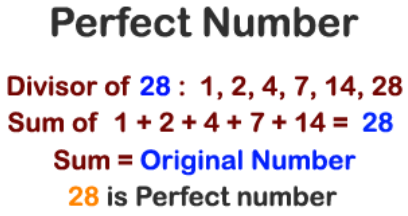
Desen şu şekildedir:




```
#include <stdio.h>  // Include the standard input/output header file.

void main() {
  int i, j, spc, rows, k;  // Declare variables 'i' and 'j' for loop counters, 'spc' for spaces, 'rows' for user input, 'k' for loop counter.

  printf("Input number of rows : ");  // Print a message to prompt user input.
  scanf("%d", &rows);  // Read the value of 'rows' from the user.
  spc = rows + 4 - 1;  // Calculate the initial number of spaces.

  for (i = 1; i <= rows; i++) {  // Start a loop to generate rows.
    for (k = spc; k >= 1; k--) {  // Loop to print spaces before the asterisks.
      printf(" ");
    }

    for (j = 1; j <= i; j++) {  // Loop to print asterisks based on the current row.
      printf("* ");  // Print an asterisk followed by a space.
    }

    printf("\n");  // Move to the next line for the next row.
    spc--;  // Decrement the number of spaces for the next row.
  }
}
```

Örnek 2: Verilen bir sayının 'Mükemmel' sayı olup olmadığını kontrol eden bir C programı yazın.

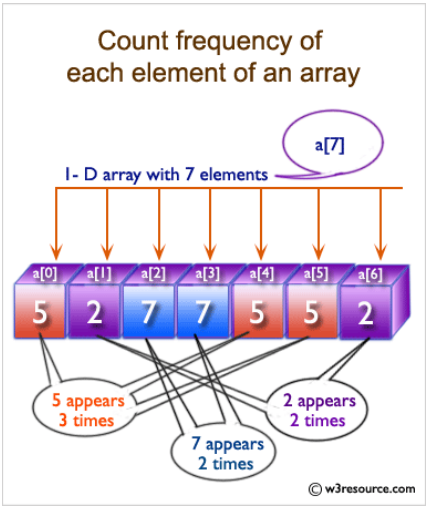
Mükemmel sayı, kendisi hariç, uygun bölenlerinin toplamına eşit olan pozitif bir tam sayıdır. Örneğin, 6 sayısı mükemmel bir sayıdır çünkü bölenleri 1, 2 ve 3'tür ve bu bölenlerin toplamı 6'dır.



```
/*  Perfect number is a positive number which sum of all positive divisors excluding that number is equal to that number.
   For example, 6 is a perfect number since the divisors of 6 are 1, 2, and 3. Sum of its divisors is 1 + 2 + 3 = 6 */

#include <stdio.h> // Include the standard input/output header file.

void  main()
{
 int n, i, sum; // Declare variables for user input, loop control, and sum.
 int mn, mx;    // Variables mn and mx are declared but not used. Consider removing them.

 printf("Input the  number : "); // Prompt the user for input.
 scanf("%d", &n); // Read the value of 'n' from the user.
 sum = 0; // Initialize the sum variable.

 printf("The positive divisors  : "); // Print a message to indicate positive divisors are being displayed.

 for (i = 1; i < n; i++) // Loop to find and display positive divisors.
 {
  if (n % i == 0) // If 'i' is a divisor of 'n'.
  {
    sum = sum + i; // Add 'i' to the sum.
    printf("%d  ", i); // Print 'i' as a positive divisor.
  }
 }

 printf("\nThe sum of the divisors is : %d", sum); // Print the sum of the divisors.

 if (sum == n) // Check if the sum of divisors is equal to the original number.
  printf("\nSo, the number is perfect."); // If true, print that the number is perfect.
 else
  printf("\nSo, the number is not perfect."); // If false, print that the number is not perfect.

 printf("\n"); // Print a new line for better formatting.
}
```

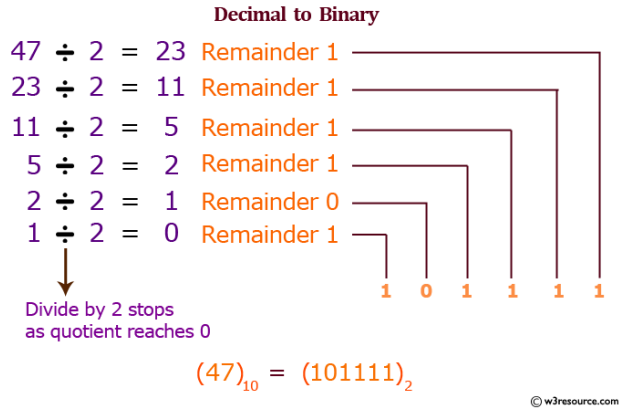
Örnek 3: Bir dizinin her bir öğesinin sıklığını saymak için C dilinde bir program yazın.

Görev, bir dizideki her bir öğenin sıklığını saymak ve görüntülemek için bir C programı yazmayı gerektirir. Program belirli sayıda tam sayı girdisi kabul edecek, bunları bir dizide saklayacak ve ardından her bir öğenin dizide kaç kez göründüğünü belirleyecek ve yazdıracaktır.



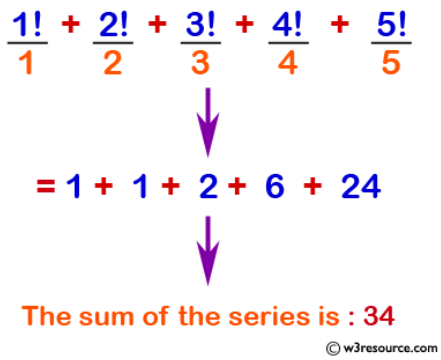
```c
#include <stdio.h>

int main()
{
   int arr1[100], fr1[100];
   int n, i, j, ctr;

   // Prompt user for input
   printf("\n\nCount frequency of each element of an array:\n");
   printf("-------------------------------------------------\n");
   printf("Input the number of elements to be stored in the array :");
   scanf("%d", &n);

   // Input elements for the array
   printf("Input %d elements in the array :\n", n);
   for (i = 0; i < n; i++)
   {
      printf("element - %d : ", i);
      scanf("%d", &arr1[i]);
      fr1[i] = -1; // Initialize frequency array with -1
   }

   // Count the frequency of each element in the array
   for (i = 0; i < n; i++)
   {
      ctr = 1; // Initialize counter for each element
      for (j = i + 1; j < n; j++)
      {
         if (arr1[i] == arr1[j])
         {
            ctr++;     // Increment counter for matching elements
            fr1[j] = 0; // Mark the duplicate element's frequency as 0
         }
      }

      // If frequency array value is not marked as 0, set it to the counter
      if (fr1[i] != 0)
      {
         fr1[i] = ctr;
      }
   }

   // Print the frequency of each element in the array
   printf("\nThe frequency of all elements of the array : \n");
   for (i = 0; i < n; i++)
   {
      if (fr1[i] != 0)
      {
         printf("%d occurs %d times\n", arr1[i], fr1[i]);
      }
          }
          return 0; }
```

Örnek 4: Fonksiyon kullanarak ondalık sayıyı ikili sayıya dönüştüren bir C programı yazın.




```
#include<stdio.h>

long toBin(int);

int main()
{
   long bno;
   int dno;
         printf("\n\n Function : convert decimal to binary :\n");
         printf("-----------------------------------------\n");
   printf(" Input any decimal number : ");
   scanf("%d",&dno);
   bno = toBin(dno);
   printf("\n The Binary value is : %ld\n\n",bno);

   return 0;
}

long toBin(int dno)
{
   long bno=0,remainder,f=1;
   while(dno != 0)
   {
      remainder = dno % 2;
      bno = bno + remainder * f;
      f = f * 10;
      dno = dno / 2;
   }
   return bno;
}
```

Örnek 5: 1!/1+2!/2+3!/3+4!/4+5!/5 serisinin toplamını fonksiyon kullanarak bulan bir C programı yazınız.



```
#include <stdio.h>

int fact(int);
void main()
{
   int sum;
   sum=fact(1)/1+fact(2)/2+fact(3)/3+fact(4)/4+fact(5)/5;
            printf("\n\n Function : find the sum of 1!/1+2!/2+3!/3+4!/4+5!/5 :\n");
            printf("-------------------------------------------------------------\n");
   printf("The sum of the series is : %d\n\n",sum);
}

int fact(int n)
   {
      int num=0,f=1;
      while(num<=n-1)
      {
         f =f+f*num;
         num++;
      }
   return f;
   }
```