1. faktöriyel sayı hesabı yapan programın kodunu recursive fonksiyon kullanarak yazınız.

|  |
| --- |
| #include <stdio.h>  #include <stdlib.h>  int faktoryel(int);  int main()  {  int n,sonuc;  printf("sayiyi giriniz\n");  scanf("%d",&n);  sonuc=faktoryel(n);  printf("sonuc=%d\n",sonuc);  return 0;  }  int faktoryel(int a)  {  if(a==1||a==0)  return 1;  else return a\*faktoryel(a-1);  } |

1. Fibonacci sayı hesabı yapan programın kodunu recursive fonksiyon kullanarak yazınız.

|  |
| --- |
| #include <stdio.h>  #include <stdlib.h>  int fib(int);  int main()  {  int n,sonuc;  printf("kacinci fibonacci sayisi\n");  scanf("%d",&n);  sonuc=fib(n);  printf("sonuc=%d\n",sonuc);  return 0;  }  int fib(int a)  {  if(a==0||a==1)  return a;  else  return fib(a-1)+fib(a-2);  } |

1. Verilen bir sayıya kadar olan tam sayıların toplamını recursive fonksiyonla bulan programın kodunu yazınız.

|  |
| --- |
| #include <stdio.h>  #include <stdlib.h>  int toplam(int);  int main()  {  int n,sonuc;  printf("sayiyi giriniz\n");  scanf("%d",&n);  sonuc=toplam(n);  printf("sonuc=%d\n",sonuc);  return 0;  }  int toplam(int a)  {  if(a==0)  return a;  else  return a+toplam(a-1);  } |

1. Bilgisayara 1 ile 3 arasında 100 rastgele üretilen her sayıdan kaç tane olduğunu bulan programın kodunu fonksiyon kullanarak yazınız.

|  |
| --- |
| #include <stdio.h>  #include <stdlib.h>  int degtopla(int[]);  int main()  {  int dizi[100],i,deg[3];  for(i=0; i<100; i++)  {  dizi[i]=rand()%3+1;  printf("%d\n",dizi[i]);  }  degtopla(dizi);  return 0;  }  int degtopla(int a[])  {  int deg[3],i;  deg[0]=0;  deg[1]=0;  deg[2]=0;  for(i=0; i<100; i++)  {  deg[a[i]-1]++;  }  for(i=0; i<3; i++)  printf("%d rakamindan %d adet vardir\n",i+1,deg[i]);  } |

1. Kullanıcının girmiş olduğu bir sayıyı ikilik tabana çeviren programın kodunu fonksiyon kullanarak yazınız.

|  |
| --- |
| #include <stdio.h>  #include <stdlib.h>  int ikiliktaban(int);  int main()  {  int sayi;  printf("sayiyi giriniz\n");  scanf("%d",&sayi);  ikiliktaban(sayi);  return 0;  }  int ikiliktaban(int a)  {  if(a>0)  {  ikiliktaban(a/2);  printf("%d",a%2);  }  } |

1. İkili arama(Binary Search) da sıralanmış elemanlar üzerinde gerçekleştirilir. Bir dizinin iki parçaya bölünmesi ve uygun parçada aynı işlemin sürdürülmesi ile yapılan arama işlemidir.

* Problemde aranacak uzayın tam orta noktasına bak
* Şayet aranan değer bulunduysa bit
* Şayet bakılan değer aranan değerden büyükse arama işlemini problem uzayının küçük elamanlarında devam ettir.
* Şayet bakılan değer arana değerden küçükse arama işlemini problem uzayının büyük elemanlarında devam ettir.
* Şayet bakılan aralık 1 veya daha küçükse aranan değer bulunamadı olarak bit

Yukardaki tanıma göre ikili aramayı yapan programın kodunu recursive fonksiyon kullanarak yazınız.

int i; int a[8] = { 1,3,4,5,17,18,31,33 };

|  |
| --- |
| #include <stdio.h>  #include <stdlib.h>  int binarysearch(int[],int,int ,int);  int main()  {  int a[8] = { 1,3,4,5,17,18,31,33 },sonuc;  sonuc=binarysearch(a,17,0,7);  if(sonuc==1)  printf("aranan deger dizi icindedir");  else  printf("aranan deger bulunamadi");  return 0;  }  int binarysearch(int a[], int aranan, int enkucuk,int enbuyuk)  {  int orta;  if(enkucuk>enbuyuk)  return -1;  orta=(enkucuk+enbuyuk)/2;  if(aranan==a[orta])  return 1;  else if(aranan<a[orta])  binarysearch(a,aranan,enkucuk,orta-1);  else if(aranan>a[orta])  binarysearch(a,aranan,orta+1,enbuyuk);  } |