**// zbir najznacajna i votra najznacajna cifra**

**int rek (int n)**

**{**

**if (n<100) return n/10 + n%10;**

**return rek(n/10);**

**}**

**// pozicija na najznacajna cifra - isto so broj an cifi**

**int rek (int n)**

**{**

**if (n<10) return 1;**

**return 1+ rek(n/10);**

**}**

**// broj na parni cifri**

**int rek (int n)**

**{**

**if (n%2 ==0)**

**{**

**if (n<10) return 1;**

**return 1+ rek(n/10);**

**}**

**else**

**{**

**if (n<10) return 0;**

**return 0+ rek(n/10);**

**}**

**}**

**// broj na neparni cifri**

**int rek (int n)**

**{**

**if (n%2 !=0)**

**{**

**if (n<10) return 1;**

**return 1+ rek(n/10);**

**}**

**else**

**{**

**if (n<10) return 0;**

**return 0+ rek(n/10);**

**}**

**}**

**// suma na cifri**

**int rek (int n)**

**{**

**if (n<10) return n;**

**return n%10 + rek(n/10);**

**}**

**// suma na parni cifri**

**int rek (int n)**

**{**

**if (n%2 ==0)**

**{**

**if (n<10) return n;**

**return n%10 + rek(n/10);**

**}**

**else**

**{**

**if (n<10) return 0;**

**return 0 + rek(n/10);**

**}**

**}**

**// suma na neparni cifri**

**int rek (int n)**

**{**

**if (n%2 !=0)**

**{**

**if (n<10) return n;**

**return n%10 + rek(n/10);**

**}**

**else**

**{**

**if (n<10) return 0;**

**return 0 + rek(n/10);**

**}**

**}**

**// suma na cifri na parni(neparni) pozicii - za neparni n/10 pred da se vnesi vo funkcijata**

**int rek (int n)**

**{**

**if (n<10) return n;**

**return n%10 + rek(n/100);**

**}**

**// proizvod na cifri na parni(neparni) pozicii - za neparni n/10 pred da se vnesi vo funkcijata**

**int rek (int n)**

**{**

**if (n<10)**

**{**

**if (n==0) return 1;**

**else return n;**

**}**

**return n%10 \* rek(n/100);**

**}**

**// faktoriel**

**int rek (int n)**

**{**

**if (n==1) return 1;**

**return n\* rek(n-1);**

**}**

**// zbir na site parni broevi od 1 do n ako n e paren broj**

**// ako n e neparen zbir na site neparni broevi od 1 do n**

**int rek (int n)**

**{**

**if (n==2) return 2;**

**if (n==1) return 1;**

**return n+ rek(n-2);**

**}**

**// proizvod na site parni broevi od 1 do n ako n e paren broj**

**// ako n e neparen proizvod na site neparni broevi od 1 do n**

**int rek (int n)**

**{**

**if (n==2) return 2;**

**if (n==1) return 1;**

**return n\* rek(n-2);**

**}**

**// prevrten broj od 123 vo 321**

**int rek (int n,int p) // p=0; vo main programata**

**{**

**if (n==0) return p;**

**return rek ( n/10 ,p\*10 + (n%10)) ;**

**}**

**// stepen**

**int rek (int m,int n) // m osnova , n stepenot**

**{**

**if (n) return m\*rek(m,n-1);**

**else return 1;**

**}**

**// prost**

**int prost(int x, int d) // d se stava 2 vo main funkcijata**

**{**

**if (x==1) return 0;**

**if (x==2) return 1;**

**if(x % d == 0) return 0;**

**if(d > x / 2) return 1;**

**return prost(x, d + 1);**

**}**

**// NZD**

**int rek ( int m, int n) // nema vrska koj e pogolem pri vnesuvanjeto vo funkcijata**

**{**

**if (!n) return m;**

**return rek (n, m % n);**

**}**

**// NZS - x segos treba da e pogolemo, y pomalo**

**int rek(int x, int y, int s) // s se stava ednakvo na pomaloto od x i y ,primer rek(x,y,y)**

**{**

**if(s % x == 0) return s;**

**return rek(x, y, s + y);**

**}**

**// dolzina na tekstualna niza rekurzivno**

**int rek(char \*p)**

**{**

**if (\*p=='\0') return 0;**

**return 1+ rek(p+1);**

**}**

**// najmal element vo niza od celobrojni broevi**

**int min\_rek (int a[], int n) // pri vnesuvanje se stava min\_rek(a,n-1) - n e brojot na elementi na nizata**

**{**

**if (n==0) return a[n] ;**

**else**

**{**

**int pom = min\_rek(a,n-1);**

**if ( a[n] < pom)**

**return a[n];**

**return pom;**

**}**

**}**

**// najgolem element vo niza od celobrojni broevi**

**int max\_rek (int a[], int n) // pri vnesuvanje se stava max\_rek(a,n-1) - n e brojot na elementi na nizata**

**{**

**if (n==0) return a[n] ;**

**else**

**{**

**int pom = max\_rek(a,n-1);**

**if ( a[n] > pom)**

**return a[n];**

**return pom;**

**}**

**}**

**// broj na pojavuvanje na dadena cifra vo celbroj ako e edna do druga duplo se broi -primerot e so cifrata 8**

**int rek (int n)**

**{**

**if (n == 0) return 0;**

**if ( (n/10) %10 == 8 && n%10 == 8 ) return 2 + rek(n/10);**

**if (n%10 == 8) return 1+ rek(n/10);**

**return rek(n/10);**

**}**

**// fibonaci**

**int rek(int n)**

**{**

**if ( n==0 || n==1 ) return n;**

**return rek(n-1) + rek(n-2);**

**}**

**// tekstualna niza -SAMO EDEN ZBOR dali e palindrom**

**int rek (char \*s, int start, int end) // se vnesuva rek (s,0,strlen(s)-1)**

**{**

**if (start >=end ) return 1;**

**if ( s[start] == s[end] )**

**return rek(s,start+1,end-1);**

**return 0;**

**}**

**// dekaden vo binaren(namesto 3 se stava 2) , ili dekaden vo tereaten(3)**

**int rek (int n)**

**{**

**if (n==0) return 0;**

**return n%3 + 10\*rek(n/3);**

**}**

**// suma na prvite N prosti broevi**

**int rek (int n)**

**{**

**if (prost(n)) // funkcija za prost broj**

**{**

**if (n<10) return n;**

**return n%10 + rek(n/10);**

**}**

**else**

**{**

**if (n<10) return 0;**

**return 0 + rek(n/10);**

**}**

**}**

**// binarno prebaruvanje rekurzivno**

**int binarno(int a[], int poc, int kraj, int key)**

**{**

**int mid;**

**mid = (poc + kraj) / 2;**

**if(a[mid] == key)**

**return mid;**

**else{**

**if(a[mid] < key) return binarno(a, mid, kraj, key);**

**else return binarno(a, 0, mid, key);**

**}**

**}**

**// podniza**