**Proje Ekip Üyeleri:** 141816044 - Burak Çayır

151816006 - Fatih Furkan Bayar

141816058 - Ramazan Uysal

141816046 - Onur Ekkazan

**Ekip Üyelerinin Görevi:**

Burak Çayır : Dönüştür , degistir\_beni , virgül\_ekle fonksiyonlarını yazmıştır.

Fatih Furkan Bayar : Sonraki , islenecekmi , geçerlimi fonksiyonlarını yazmıştır.

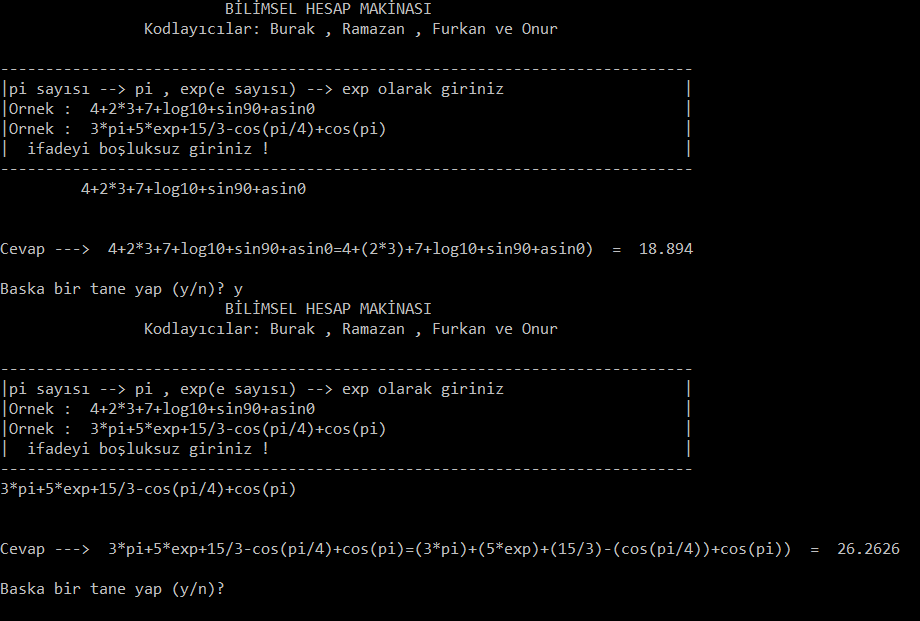
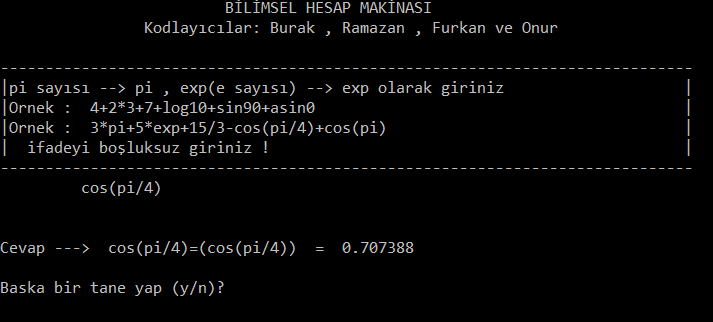
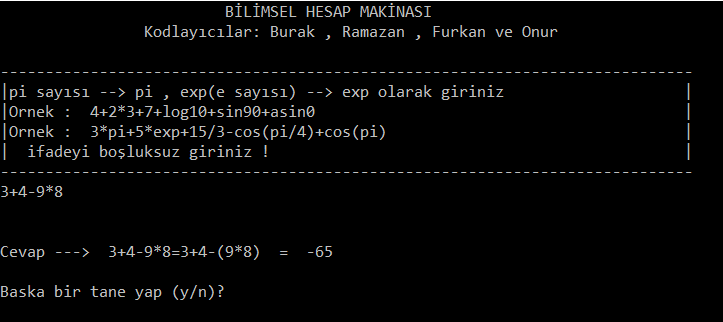
Ramazan Uysal : oncelik , parantez ve cevir fonksiyonlarını yazmıştır.

Onur Ekkazan : Hesapla , fakt , sgn fonksiyonlarını yazmıştır.

**Kullanılan Programlama Dili:** C++

**Kullanılan Veri Yapısı:** Stack

**Ekran Çıktıları:**



**Genel Çalışma Mantığı:**

Programızın kullanıcı dostu bir arayüzü vardır. Programımız kullanıcıdan bir işlem alır. Daha sonra bazı işlemleri (trigonometrik vb.) programın anlayacağı şekle çevirir. Ardından programımız kullanıcının girdiği ifadenin matematiksel uygunluğunu kontrol eder. Eğer geçerli değilse “hata” mesajı yansıtılır. Eğer geçerli ise ifadeyi Stack veri yapısı kullanarak postfix ifadeye dönüştürür. Hesaplama yapılır ve sonra öncelikli işlemleri belirtme amacıyla parantezler eklenir. Ardından kullanıcının anlayacağı şekilde ifade ekrana yansıtılır.

**Kodlar:**

**#include <iostream>**

**#include <sstream>**

**#include <string>**

**#include <stack>**

**#include <vector>**

**#include <math.h>**

**#include <cmath>**

**#include <locale.h>**

**//Gerekli Kütüphaneler Eklendi**

**using namespace std;**

**//fonksiyon protipleri(imzalari)**

**void donustur(const string & Infix, string & Postfix);//BURAK**

**void Sonraki(string);//FURKAN**

**bool islenecekmi(char ch);//FURKAN**

**bool oncelik(char OperatorA, char OperatorB);//RAMAZAN**

**bool gecerlimi(string);//FURKAN**

**double Hesapla(string[]);//ONUR**

**double fakt(double);//ONUR**

**double sgn(double);//ONUR**

**string degistir\_beni(string); //BURAK**

**string virgul\_ekle(string); //BURAK**

**string parantez(string);// RAMAZAN**

**string cevir(string); //RAMAZAN**

**//Main Fonksiyonu**

**int main()**

**{**

**setlocale(LC\_ALL,"Turkish");**

**char Cevapla;//while Döngüsü İcin Degisken**

**do**

**{**

**string Infix , Infix2 , Postfix;**

**cout <<"\t\t\t BİLİMSEL HESAP MAKİNASI \n";**

**cout <<"\t\tKodlayıcılar: Burak , Ramazan , Furkan ve Onur\n\n";**

**cout <<"-----------------------------------------------------------------------------\n";**

**cout <<"|pi sayısı --> pi , exp(e sayısı) --> exp olarak giriniz |\n";**

**cout <<"|Ornek : 4+2\*3+7+log10+sin90+asin0 |\n";**

**cout <<"|Ornek : 3\*pi+5\*exp+15/3-cos(pi/4)+cos(pi) |\n";**

**cout <<"| ifadeyi boşluksuz giriniz ! |\n";**

**cout <<"-----------------------------------------------------------------------------\n";**

**string orjinal;**

**cin>>orjinal;**

**Infix=cevir(orjinal); //makinanın anlayacağı dile çeviriyoruz.**

**if(gecerlimi(Infix)==true)**

**{**

**string temp;**

**temp = degistir\_beni(Infix);**

**donustur(temp, Postfix);**

**string tut;**

**tut = virgul\_ekle(Postfix);**

**Infix2 = parantez(orjinal);**

**cout<<"\n\nCevap ---> "<<orjinal<<"="<<Infix2<<" = "; //kullanıcıya anlayacağı şekilde gösteriyoruz.**

**Sonraki(tut);**

**cout << endl << "\nBaska bir tane yap (y/n)? ";**

**cin >> Cevapla;**

**}**

**else**

**{**

**cout<<"\*\*\*HATA\*\*\*\n";**

**cout << endl << "Tekrar yap (y/n)? ";**

**cin >> Cevapla;**

**}**

**}while (tolower(Cevapla) == 'y');**

**return 0;**

**}**

**string cevir(string x)**

**{**

**int i=0;**

**for(i=0 ; i<x.length() ; i++)**

**{**

**if(x[i]=='c' && x[i+1]=='o' && x[i+2]=='s') //cos için**

**{**

**x[i]='1';**

**x[i+1]='c';**

**x.erase(i+2,1);**

**}**

**if(x[i]=='s' && x[i+1]=='i' && x[i+2]=='n') //sin için**

**{**

**x[i]='1';**

**x[i+1]='s';**

**x.erase(i+2,1);**

**}**

**if(x[i]=='t' && x[i+1]=='a' && x[i+2]=='n') //tan için**

**{**

**x[i]='1';**

**x[i+1]='t';**

**x.erase(i+2,1);**

**}**

**if(x[i]=='c' && x[i+1]=='o' && x[i+2]=='t') //cot için**

**{**

**x[i]='1';**

**x[i+1]='a';**

**x.erase(i+2,1);**

**}**

**if(x[i]=='l' && x[i+1]=='o' && x[i+2]=='g') //log için**

**{**

**x[i]='1';**

**x[i+1]='h';**

**x.erase(i+2,1);**

**}**

**if(x[i]=='e' && x[i+1]=='x' && x[i+2]=='p') //exp için**

**{**

**x[i]='1';**

**x[i+1]='k';**

**x[i+2]='1';**

**}**

**if(x[i]=='a' && x[i+1]=='s' && x[i+2]=='i' && x[i+3]=='n') //asin için**

**{**

**x[i]='1';**

**x[i+1]='b';**

**x.erase(i+2,2);**

**}**

**if(x[i]=='a' && x[i+1]=='c' && x[i+2]=='o' && x[i+3]=='s') //acos için**

**{**

**x[i]='1';**

**x[i+1]='d';**

**x.erase(i+2,2);**

**}**

**if(x[i]=='a' && x[i+1]=='t' && x[i+2]=='a' && x[i+3]=='n') //atan için**

**{**

**x[i]='1';**

**x[i+1]='e';**

**x.erase(i+2,2);**

**}**

**if(x[i]=='a' && x[i+1]=='c' && x[i+2]=='o' && x[i+3]=='t') //acot için**

**{**

**x[i]='1';**

**x[i+1]='f';**

**x.erase(i+2,2);**

**}**

**if(x[i]=='l' && x[i+1]=='n') //ln için**

**{**

**x[i]='1';**

**x[i+1]='g';**

**}**

**if(x[i]=='p' && x[i+1]=='i') //PI için**

**{**

**x[i]='1';**

**x[i+1]='j';**

**x.insert(i+2,1,'1');**

**}**

**if(x[i]=='!') //! (faktöriyel) için**

**{**

**x[i]='!';**

**x.insert(i+1,1,'1');**

**}**

**}**

**return x;**

**}**

**//Parentezli Yazdirmak İcin Fonksiyon //String İfade döndürür**

**string parantez(string x)**

**{**

**string burak2;**

**x="::"+x+"::";**

**int sayac=0,i,j;**

**int uzunluk = x.length();**

**for(i = 1; i<=uzunluk; i++)**

**{**

**if(x[i]=='\*' || x[i]=='/')**

**{**

**for(j=i; j>0; j--)**

**{**

**if(x[j]=='+' || x[j]=='-' || x[j]==':')**

**{**

**x.insert(j+1,1,'(');**

**i=i+1;**

**break;**

**}**

**}**

**for(j=i; j<uzunluk; j++)**

**{**

**if(x[j]=='+' || x[j]=='-')**

**{**

**x.insert(j,1,')');**

**i=i+1;**

**break;**

**}**

**}**

**for(j=i; j<uzunluk; j++)**

**{**

**if (x[j+1]==':')**

**{**

**x.insert(j+1,1,')');**

**break;**

**}**

**}**

**}**

**}**

**string gecici;**

**for(i=2 ; i<x.length() -2 ; i++)**

**{**

**gecici += x[i];**

**}**

**return gecici;**

**}**

**//Faktöriyel Fonksiyonu**

**double fakt(double x)**

**{**

**int faktoriyel=1;**

**for(int i=1; i<x; i++){**

**faktoriyel = faktoriyel + faktoriyel\*i;**

**}**

**return faktoriyel;**

**}**

**//Sgn(İsaret) Fonksiyonu**

**double sgn(double x)**

**{**

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

**if(x<0) return -1;**

**return 0;**

**}**

**//Return: eğer ch bir operatorse false , değilse true**

**bool islenecekmi(char ch)**

**{**

**if (**

**((ch >= 'l') && (ch < 's')) ||**

**((ch >= 'u') && (ch < 'z')) ||**

**((ch >= '0') && (ch <= '9'))**

**)**

**return true;**

**else**

**return false;**

**}**

**/\*OperatorA operatör yada parantez olacak.**

**OperatorB operatör yada parantez olacak.**

**Görev: OperatorA OperatorB den yüksek öncelikli durumları belirle**

**Return: OperatorA OperatorB den yüksek öncelikli durumda ise true döndür\*/**

**bool oncelik(char OperatorA, char OperatorB)**

**{**

**if (OperatorA == '(')**

**return false;**

**else if (OperatorB == '(')**

**return false;**

**else if (OperatorB == ')')**

**return true;**

**else if ((OperatorA == '^') && (OperatorB == '^'))**

**return false;**

**else if (OperatorA == '^')**

**return true;**

**else if (OperatorB == '^')**

**return false;**

**else if ((OperatorA == '%') && (OperatorB == '%'))**

**return false;**

**else if (OperatorA == '%')**

**return true;**

**else if (OperatorB == '%')**

**return false;**

**else if ((OperatorA == '!') && (OperatorB == '!'))**

**return false;**

**else if (OperatorA == '!')**

**return true;**

**else if (OperatorB == '!')**

**return false;**

**else if ((OperatorA == 'a') && (OperatorB == 'a'))**

**return false;**

**else if (OperatorA == 'a')**

**return true;**

**else if (OperatorB == 'a')**

**return false;**

**else if ((OperatorA == 'b') && (OperatorB == 'b'))**

**return false;**

**else if (OperatorA == 'b')**

**return true;**

**else if (OperatorB == 'b')**

**return false;**

**else if ((OperatorA == 's') && (OperatorB == 's'))**

**return false;**

**else if (OperatorA == 's')**

**return true;**

**else if (OperatorB == 's')**

**return false;**

**else if ((OperatorA == 't') && (OperatorB == 't'))**

**return false;**

**else if (OperatorA == 't')**

**return true;**

**else if (OperatorB == 't')**

**return false;**

**else if ((OperatorA == 'c') && (OperatorB == 'c'))**

**return false;**

**else if (OperatorA == 'c')**

**return true;**

**else if (OperatorB == 'c')**

**return false;**

**else if ((OperatorA == 'd') && (OperatorB == 'd'))**

**return false;**

**else if (OperatorA == 'd')**

**return true;**

**else if (OperatorB == 'd')**

**return false;**

**else if ((OperatorA == 'e') && (OperatorB == 'e'))**

**return false;**

**else if (OperatorA == 'e')**

**return true;**

**else if (OperatorB == 'e')**

**return false;**

**else if ((OperatorA == 'f') && (OperatorB == 'f'))**

**return false;**

**else if (OperatorA == 'f')**

**return true;**

**else if (OperatorB == 'f')**

**return false;**

**else if ((OperatorA == 'g') && (OperatorB == 'g'))**

**return false;**

**else if (OperatorA == 'g')**

**return true;**

**else if (OperatorB == 'g')**

**return false;**

**else if ((OperatorA == 'h') && (OperatorB == 'h'))**

**return false;**

**else if (OperatorA == 'h')**

**return true;**

**else if (OperatorB == 'h')**

**return false;**

**else if ((OperatorA == 'i') && (OperatorB == 'i'))**

**return false;**

**else if (OperatorA == 'i')**

**return true;**

**else if (OperatorB == 'i')**

**return false;**

**else if ((OperatorA == 'j') && (OperatorB == 'j'))**

**return false;**

**else if (OperatorA == 'j')**

**return true;**

**else if (OperatorB == 'j')**

**return false;**

**else if ((OperatorA == 'k') && (OperatorB == 'k'))**

**return false;**

**else if (OperatorA == 'k')**

**return true;**

**else if (OperatorB == 'k')**

**return false;**

**else if ((OperatorA == '\*') || (OperatorA == '/'))**

**return true;**

**else if ((OperatorB == '\*') || (OperatorB == '/'))**

**return false;**

**else**

**return true;**

**}**

**/\* Verilen: Infix infix ifadeyi temsil eden bir string ifadesi (boşluksuz).**

**Görev: infix ifadenin postfix eşdeğerini bul.**

**Return: Postfix ifadeyi tutan bir string döndür .**

**\*/**

**void donustur(const string & Infix, string & Postfix)**

**{**

**stack<char> OperatorStack;//Char Türünde Oparatör icin Stack Tanimlandi**

**char en\_yukardaki\_sembol, sembol;**

**int k;**

**for (k = 0; k < Infix.size(); k++)**

**{**

**sembol = Infix[k];**

**if (islenecekmi(sembol))**

**Postfix = Postfix + sembol;//eğer karakterimiz operatör değil veya sayı ise postfixe ekleniyor.**

**else**

**{//operatör ise**

**while (**

**(! OperatorStack.empty()) && //eğer stack boş değil ve**

**(oncelik(OperatorStack.top(), sembol))//stackin üstündeki elemanın girilen karaktere önceliği varsa**

**)**

**{**

**en\_yukardaki\_sembol = OperatorStack.top(); //stackin son elemanı bu degiskene atanıyor**

**OperatorStack.pop(); //en yukardaki eleman siliniyor**

**Postfix = Postfix + en\_yukardaki\_sembol; //sonra bu eleman postfix'in en sonuna ekleniyor**

**} //while bitti**

**if ((! OperatorStack.empty()) && (sembol == ')')) //eğer stack boş değil ve sembol==')' ise**

**OperatorStack.pop(); // bu eşleşmeyi çıkart**

**else**

**OperatorStack.push(sembol); //ne olursa olsun stacke ekle**

**}**

**}**

**while (! OperatorStack.empty()) //stack boş değilse**

**{**

**en\_yukardaki\_sembol = OperatorStack.top();**

**OperatorStack.pop();**

**Postfix = Postfix + en\_yukardaki\_sembol;**

**}**

**}**

**//degistir fonksiyonu //String deger döndürür**

**string degistir\_beni(string stringim)**

**{**

**for(int i = 0; i <stringim.length(); i++)**

**{**

**if(isdigit(stringim[i])!=0) //eğer ch dijitse (0,1,2,3,4,5,6,7,8,9)**

**{**

**if(isdigit(stringim[i+1])==0) //eger ch digit değilse**

**{**

**stringim.insert(i+1, "v"); //ör : 1v2 oldu**

**//v burada keyfi bir seçimdir**

**//ama bir harf olmalı**

**}**

**}**

**}**

**// -7\*-7 durumunda değiştir Nedeni satır 487 den anlaşılacaktır. (burayı şimdilik atla)**

**for (int i = 0; i <stringim.length(); i++)**

**{**

**if(stringim[i]=='-')**

**{**

**if((stringim[i-1]!='v')&&(stringim[i-1]!=')')) //- den önceki v veya ) değilse**

**{**

**stringim.replace(i,1,"y"); // "-" yi "y" ile degistir.**

**}**

**}**

**}**

**return stringim;**

**}**

**//Infıx Posfıx yazilimda Virgül ekler //String ifade döndürür**

**string virgul\_ekle(string stringim)**

**{**

**for(int i = 0; i <stringim.length(); i++)**

**{**

**if(**

**//operatörler için aralara virgül ekleyeceğiz.**

**(stringim[i]=='\*')||**

**(stringim[i]=='-')||**

**(stringim[i]=='/')||**

**(stringim[i]=='+')||**

**(stringim[i]=='^')||**

**(stringim[i]=='%')||**

**(stringim[i]=='!')||**

**(stringim[i]=='s')||**

**(stringim[i]=='c')||**

**(stringim[i]=='a')||**

**(stringim[i]=='b')||**

**(stringim[i]=='d')||**

**(stringim[i]=='e')||**

**(stringim[i]=='f')||**

**(stringim[i]=='t')||**

**(stringim[i]=='g')||**

**(stringim[i]=='h')||**

**(stringim[i]=='i')||**

**(stringim[i]=='j')||**

**(stringim[i]=='k')**

**)**

**{**

**stringim.insert(i+1, ",");**

**//yukardaki operatorleri bulduysan bir virgül ekle**

**}**

**else if(stringim[i]=='v')**

**{**

**stringim.replace(i,1,","); //bu şekilde digitle operatör arasına virgül ekledik.**

**//v yi virgülle değiştir**

**}**

**}**

**//degistir**

**for (int i = 0; i <stringim.length(); i++)**

**{**

**if(stringim[i]=='y')**

**{**

**stringim.replace(i,1,"-"); //çünkü - olarak kalsaydı postfixe çevirirken sıkıntı yaşayabilirdik.**

**}**

**}**

**return stringim;**

**}**

**//fonksiyon ifadenin değişken olup olmadığını kontrol ediyor.True/False İfade döndürür**

**bool gecerlimi(string stringim)**

**{**

**//arka arkaya '+' '-' gelirse**

**for (int i = 0; i<stringim.length(); i++)**

**{**

**if((stringim[i]=='+')||(stringim[i]=='-'))**

**{**

**if((stringim[i+1]=='+')||(stringim[i+1]=='-'))**

**{**

**return false;**

**}**

**}**

**}**

**string dizi="0123456789+-\*/()^%!sctabdefghijk";**

**int saydir=0;**

**for (int i=0; i<stringim.length(); i++)**

**{**

**for(int j=0; j<dizi.length(); j++)**

**{**

**if(stringim[i]==dizi[j])**

**{**

**saydir++;**

**}**

**}**

**}**

**if (saydir == stringim.length())**

**{**

**return true;**

**}**

**else**

**{**

**return false;**

**}**

**}**

**//postfix ifadeyi hesaplıyor**

**void Sonraki(string stringim)**

**{**

**vector <string> dizi;**

**string tempy;**

**int virgul\_saydir=0;**

**for (int a=0; a<stringim.length();a++)**

**{**

**if(stringim[a]==',')**

**{**

**virgul\_saydir++;**

**}**

**}**

**//"," bir sınırlayıcı olarak kullanıyoruz**

**while (stringim.find(",", 0) != string::npos)//string sonuna kadar**

**{**

**//string virgül içeriyormu**

**size\_t pos = stringim.find(",", 0);**

**tempy = stringim.substr(0, pos);**

**stringim.erase(0, pos + 1);**

**dizi.push\_back(tempy); //vektörde alışveriş et**

**}**

**stack <string> my\_stack;**

**string temp[100];**

**string ch;**

**for (int i=0; i<virgul\_saydir; i++)**

**{**

**string s;**

**s=dizi[i];**

**if (**

**(s!="+")&&**

**(s!="\*")&&**

**(s!="-")&&**

**(s!="/")&&**

**(s!="^")&&**

**(s!="%")&&**

**(s!="!")&&**

**(s!="s")&&**

**(s!="c")&&**

**(s!="t")&&**

**(s!="a")&&**

**(s!="b")&&**

**(s!="d")&&**

**(s!="e")&&**

**(s!="f")&&**

**(s!="g")&&**

**(s!="h")&&**

**(s!="i")&&**

**(s!="j")&&**

**(s!="k")**

**)**

**{**

**my\_stack.push(s);**

**//stack içine push et**

**}**

**else //eğer bir operatorle karşılaşırsa**

**{**

**my\_stack.push(s);//stack e push et**

**temp[0]= my\_stack.top();//degeri al**

**my\_stack.pop(); //stack den sil**

**temp[1]= my\_stack.top();//degeri al**

**my\_stack.pop();//stack den sil**

**temp[2]= my\_stack.top();//degeri al**

**my\_stack.pop();//stack den sil**

**double z;**

**z = Hesapla(temp);//Hesapla Fonksiyonu Cagriliyor**

**ostringstream outs; // stringstream olarak tanımla**

**outs << z; // stringe dönüştü.**

**ch = outs.str();**

**my\_stack.push(ch);//My\_stack ch yi ekliyor**

**}**

**}**

**cout<<ch;**

**cin.get();**

**}**

**//matematiksel işlemler: önce double a sonra tekrar stringe donuşturulecekler**

**double Hesapla(string temp[])**

**{**

**string a,b,c;**

**a=temp[2]; b=temp[0]; c=temp[1];**

**double x,y,z;**

**istringstream ins,inse;**

**ins.str(a);inse.str(c);**

**ins >> x;**

**inse >> y;**

**if (b=="+")//Toplama**

**{**

**z = x + y;**

**return z;**

**}**

**else if (b=="-")//Cikarma**

**{**

**z = x - y;**

**return z;**

**}**

**else if (b=="\*")//Carpma**

**{**

**z = x \* y;**

**return z;**

**}**

**else if (b=="/")//Bölme**

**{**

**z = x / y;**

**return z;**

**}**

**else if (b=="^")//Üs Alma**

**{**

**z = pow(x,y);**

**return z;**

**}**

**else if (b=="%")//Mod Alma**

**{**

**z = (int)x % (int)y;**

**return z;**

**}**

**else if (b=="!")//Faktöriyel**

**{**

**z = fakt(x);//fakt fonksiyonu Cagriliyor**

**return z;**

**}**

**else if (b=="s")//Sin**

**{**

**z = sin(y);**

**return z;**

**}**

**else if (b=="c")//Cos**

**{**

**z = cos(y);**

**return z;**

**}**

**else if (b=="t")//Tan**

**{**

**z = tan(y);**

**return z;**

**}**

**else if (b=="a") //cot**

**{**

**z = (cos(y))/(sin(y));**

**return z;**

**}**

**else if (b=="b") //asin**

**{**

**z = asin(y);**

**return z;**

**}**

**else if (b=="d") //acos**

**{**

**z = acos(y);**

**return z;**

**}**

**else if (b=="e") //atan**

**{**

**z = atan(y);**

**return z;**

**}**

**else if (b=="f") //acot**

**{**

**z = 1/atan(y);**

**return z;**

**}**

**else if (b=="g") //ln//Logaritma**

**{**

**z = log(y);**

**return z;**

**}**

**else if (b=="h") //ln//Dogal Logaritma**

**{**

**z = log10(y);**

**return z;**

**}**

**else if (b=="i") //İsaret Fonksiyonu**

**{**

**z = sgn(y);//Sgn Fonksiyonu Cagriliyor**

**return z;**

**}**

**else if (b=="j") //Pİ**

**{**

**z = 3.14;**

**return z;**

**}**

**else if (b=="k") //e**

**{**

**z = 2.71;**

**return z;**

**}**

**}**