

PARSER MADE ENTIRELY BY US

```
#include<iostream>
#include<cmath>
#include<cstring>
#include<cstdio>

using namespace std;

char fun[60];
double X;
int i=0,n;

double mult(double x,int pow){
double z1=1;
for(int j=0;j<pow;j++){
    z1*=x;
}
return z1;
}

//The Parsing Function...
//It returns the value of the function...

double valf(double &x){

double z=1.0;
int l;
l=strlen(fun);

if (fun[i]=='\0'){return -1;}

    if((fun[i]>='0')&&(fun[i]<='9')){z=fun[i]-48;
    switch(fun[i+1]){

        case '*':{i+=2;return z*valf(x);}
        case '+':{i+=2;return z+valf(x);}
        case '-':{i+=2;return z-valf(x);}
        case '/':{i+=2;return z/valf(x);}
        case '\0':{i=0;return z;}
        default :{return -1000;}}}

if((fun[i]=='x')&&((fun[i+1]!='^')||(fun[i+1]=='')||(fun[i+1]=='\0'))){s
witch(fun[i+1]){
    case '*':{i+=2;return x*valf(x);}
    case '+':{i+=2;return x+valf(x);}
    case '-':{i+=2;return x-valf(x);}
    case '/':{i+=2;return x/valf(x);}
    case ')':{i+=1;return x;}
    case '\0':{i=0;return x;}
    default :{return -1000;}}}
```

```

if ((fun[i]=='x') && (fun[i+1]=='^')) {n=fun[i+2]-48; z=mult(x,n);
    switch(fun[i+3]) {
        case '*': {i+=4; return z*valf(x); break;}
        case '+': {i+=4; return z+valf(x); break;}
        case '-': {i+=4; return z-valf(x); break;}
        case '/': {i+=4; return z/valf(x); break;}
        case '\0': {i=0; return z; break;}
        default : {return -1000;}
    }
}

```

```

if ((fun[i]=='s') && (fun[i+1]=='i') &&
    (fun[i+2]=='n') && (fun[i+3]=='('))
    {i+=4; z=sin(valf(x));
    switch(fun[i+1]) {
        case '*': {i+=2; return z*valf(x);}
        case '+': {i+=2; return z+valf(x);}
        case '-': {i+=2; return z-valf(x);}
        case '/': {i+=2; return z/valf(x);}
        case ')': {i+=1; return z;}
        case '\0': {i=0; return z;}
        default : {return -1000;}
    }}

```

```

if ((fun[i]=='s') && (fun[i+1]=='i') &&
    (fun[i+2]=='n') && (fun[i+3]=='(') &&
    (fun[i+4]=='x') && (fun[i+5]=='(') &&
    (fun[i+6]=='^'))
    {n=fun[i+7]-48; z=mult(sin(x),n);
        switch(fun[i+8]) {
        case '*': {i+=9; return z*valf(x);}
        case '+': {i+=9; return z+valf(x);}
        case '-': {i+=9; return z-valf(x);}
        case '/': {i+=9; return z/valf(x);}
        case '\0': {i=0; return z;}
        default : {return -1000;}
        }}

```

```

if ((fun[i]=='c') && (fun[i+1]=='o') &&
    (fun[i+2]=='s') && (fun[i+3]=='('))
    {i+=4; z=cos(valf(x));
    switch(fun[i+1]) {
        case '*': {i+=2; return z*valf(x);}
        case '+': {i+=2; return z+valf(x);}
        case '-': {i+=2; return z-valf(x);}
        case '/': {i+=2; return z/valf(x);}
        case ')': {i+=1; return z;}
        case '\0': {i=0; return z;}
        default : {return -1000;}
    }}

```

```

if ((fun[i]=='c') && (fun[i+1]=='o') &&
    (fun[i+2]=='s') && (fun[i+3]=='(') &&
    (fun[i+4]=='x') && (fun[i+5]=='(') &&
    (fun[i+6]=='^'))

```

```

        {n=fun[i+7]-48;z=mult(cos(x),n);
          switch(fun[i+8]){
case '*':{i+=9;return z*valf(x);}
case '+':{i+=9;return z+valf(x);}
case '-':{i+=9;return z-valf(x);}
case '/':{i+=9;return z/valf(x);}
case '\0':{i=0;return z;}
default :{return -1000;}
}}

if((fun[i]=='t')&&(fun[i+1]=='a')&&
    (fun[i+2]=='n')&&(fun[i+3]=='(')
    ){i+=4;z=tan(valf(x));
    switch(fun[i+1]){
case '*':{i+=2;return z*valf(x);}
case '+':{i+=2;return z+valf(x);}
case '-':{i+=2;return z-valf(x);}
case '/':{i+=2;return z/valf(x);}
case ')':{i+=1;return z;}
case '\0':{i=0;return z;}
default :{return -1000;}
}}

if((fun[i]=='t')&&(fun[i+1]=='a')&&
    (fun[i+2]=='n')&&(fun[i+3]=='(')&&
    (fun[i+4]=='x')&&(fun[i+5]=='')&&
    (fun[i+6]=='^'))
    {n=fun[i+7]-48;z=mult(tan(x),n);
      switch(fun[i+8]){
case '*':{i+=9;return z*valf(x);}
case '+':{i+=9;return z+valf(x);}
case '-':{i+=9;return z-valf(x);}
case '/':{i+=9;return z/valf(x);}
case '\0':{i=0;return z;}
default :{return -1000;}
}}

if((fun[i]=='l')&&(fun[i+1]=='o')&&
    (fun[i+2]=='g')&&(fun[i+3]=='(')
    ){i+=4;z=log(valf(x));
    switch(fun[i+1]){
case '*':{i+=2;return z*valf(x);}
case '+':{i+=2;return z+valf(x);}
case '-':{i+=2;return z-valf(x);}
case '/':{i+=2;return z/valf(x);}
case ')':{i+=1;return z;}
case '\0':{i=0;return z;}
default :{return -1000;}
}}

if((fun[i]=='l')&&(fun[i+1]=='o')&&
    (fun[i+2]=='g')&&(fun[i+3]=='(')&&
    (fun[i+4]=='x')&&(fun[i+5]=='')&&
    (fun[i+6]=='^'))
    {n=fun[i+7]-48;z=mult(log(x),n);
      switch(fun[i+8]){
case '*':{i+=9;return z*valf(x);}
case '+':{i+=9;return z+valf(x);}

```

```

        case '-':{i+=9;return z-Valf(x);}
        case '/':{i+=9;return z/Valf(x);}
        case '\0':{i=0;return z;}
        default :{return -1000;}
    }}

if((fun[i]=='e') && (fun[i+1]=='x') &&
    (fun[i+2]=='p') && (fun[i+3]=='(')
    ){i+=4;z=exp(Valf(x));
    switch(fun[i+1]){
        case '*':{i+=2;return z*Valf(x);}
        case '+':{i+=2;return z+Valf(x);}
        case '-':{i+=2;return z-Valf(x);}
        case '/':{i+=2;return z/Valf(x);}
        case ')':{i+=1;return z;}
        case '\0':{i=0;return z;}
        default :{return -1000;}
    }}

if((fun[i]=='e') && (fun[i+1]=='x') &&
    (fun[i+2]=='p') && (fun[i+3]=='(') &&
    (fun[i+4]=='x') && (fun[i+5]=='') &&
    (fun[i+6]=='^'))
    {n=fun[i+7]-48;z=mult(exp(x),n);
    switch(fun[i+8]){
case '*':{i+=9;return z*Valf(x);}
        case '+':{i+=9;return z+Valf(x);}
        case '-':{i+=9;return z-Valf(x);}
        case '/':{i+=9;return z/Valf(x);}
        case '\0':{i=0;return z;}
        default :{return -1000;}
    }}

if((fun[i]=='s') && (fun[i+1]=='e') &&
    (fun[i+2]=='c') && (fun[i+3]=='(')
    ){i+=4;z=cos(Valf(x));
    switch(fun[i+1]){
        case '*':{i+=2;return (1/z)*Valf(x);}
        case '+':{i+=2;return (1/z)+Valf(x);}
        case '-':{i+=2;return (1/z)-Valf(x);}
        case '/':{i+=2;return (1/z)/Valf(x);}
        case ')':{i+=1;return (1/z);}
        case '\0':{i=0;return (1/z);}
        default :{return -1000;}
    }}

if((fun[i]=='c') && (fun[i+1]=='o') &&
    (fun[i+2]=='t') && (fun[i+3]=='(')
    ){i+=4;z=1/tan(Valf(x));
    switch(fun[i+1]){
        case '*':{i+=2;return z*Valf(x);}
        case '+':{i+=2;return z+Valf(x);}
        case '-':{i+=2;return z-Valf(x);}
        case '/':{i+=2;return z/Valf(x);}
        case ')':{i+=1;return z;}
        case '\0':{i=0;return z;}
    }}

```

```

        default :{return -1000;}
    }}

    if((fun[i]=='c')&&(fun[i+1]=='o')&&
        (fun[i+2]=='s')&&(fun[i+3]=='e')&&(fun[i+4]=='c')&&(fun[i+5]=='(')
        ){i+=6;z=sin(valf(x));
        switch(fun[i+1]){
            case '*':{i+=2;return (1/z)*valf(x);}
            case '+':{i+=2;return (1/z)+valf(x);}
            case '-':{i+=2;return (1/z)-valf(x);}
            case '/':{i+=2;return (1/z)/valf(x);}
            case ')':{i+=1;return (1/z);}
            case '\\0':{i=0;return (1/z);}
            default :{return -1000;}
        }}

    if((fun[i]=='s')&&(fun[i+1]=='q')&&
        (fun[i+2]=='r')&&(fun[i+3]=='t')&&(fun[i+4]=='(')
        ){i+=5;z=sqrt(valf(x));
        switch(fun[i+1]){
            case '*':{i+=2;return z*valf(x);}
            case '+':{i+=2;return z+valf(x);}
            case '-':{i+=2;return z-valf(x);}
            case '/':{i+=2;return z/valf(x);}
            case ')':{i+=1;return z;}
            case '\\0':{i=0;return z;}
            default :{return -1000;}
        }}

    if((fun[i]=='s')&&(fun[i+1]=='i')&&
        (fun[i+2]=='n')&&(fun[i+3]=='h')&&(fun[i+4]=='(')
        ){i+=5;z=sinh(valf(x));
        switch(fun[i+1]){
            case '*':{i+=2;return z*valf(x);}
            case '+':{i+=2;return z+valf(x);}
            case '-':{i+=2;return z-valf(x);}
            case '/':{i+=2;return z/valf(x);}
            case ')':{i+=1;return z;}
            case '\\0':{i=0;return z;}
            default :{return -1000;}
        }}

    if((fun[i]=='s')&&(fun[i+1]=='i')&&
        (fun[i+2]=='n')&&(fun[i+3]=='h')&&
        (fun[i+4]=='(')&&(fun[i+5]=='x')&&
        (fun[i+6]==')')&&(fun[i+7]=='^'))
        {n=fun[i+8]-48;z=mult(sinh(x),n);
        switch(fun[i+9]){
            case '*':{i+=10;return z*valf(x);}
            case '+':{i+=10;return z+valf(x);}
            case '-':{i+=10;return z-valf(x);}
            case '/':{i+=10;return z/valf(x);}
            case '\\0':{i=10;return z;}
            default :{return -1000;}
        }}

```

```

if((fun[i]=='c') && (fun[i+1]=='o') &&
    (fun[i+2]=='s') && (fun[i+3]=='h') && (fun[i+4]=='(')
    ){i+=5;z=cosh(valf(x));
switch(fun[i+1]){
case '*':{i+=2;return z*valf(x);}
case '+':{i+=2;return z+valf(x);}
case '-':{i+=2;return z-valf(x);}
case '/':{i+=2;return z/valf(x);}
case ')':{i+=1;return z;}
case '\\0':{i=0;return z;}
default :{return -1000;}
}}

```

```

if((fun[i]=='c') && (fun[i+1]=='o') &&
    (fun[i+2]=='s') && (fun[i+3]=='h') &&
    (fun[i+4]=='(') && (fun[i+5]=='x') &&
    (fun[i+6]==')') && (fun[i+7]=='^'))
    {n=fun[i+8]-48;z=mult(cosh(x),n);
    switch(fun[i+9]){
case '*':{i+=10;return z*valf(x);}
case '+':{i+=10;return z+valf(x);}
case '-':{i+=10;return z-valf(x);}
case '/':{i+=10;return z/valf(x);}
case '\\0':{i=10;return z;}
default :{return -1000;}
}}

```

```

if((fun[i]=='t') && (fun[i+1]=='a') &&
    (fun[i+2]=='n') && (fun[i+3]=='h') && (fun[i+4]=='(')
    ){i+=5;z=tanh(valf(x));
switch(fun[i+1]){
case '*':{i+=2;return z*valf(x);}
case '+':{i+=2;return z+valf(x);}
case '-':{i+=2;return z-valf(x);}
case '/':{i+=2;return z/valf(x);}
case ')':{i+=1;return z;}
case '\\0':{i=0;return z;}
default :{return -1000;}
}}

```

```

if((fun[i]=='t') && (fun[i+1]=='a') &&
    (fun[i+2]=='n') && (fun[i+3]=='h') &&
    (fun[i+4]=='(') && (fun[i+5]=='x') &&
    (fun[i+6]==')') && (fun[i+7]=='^'))
    {n=fun[i+8]-48;z=mult(tanh(x),n);
    switch(fun[i+9]){
case '*':{i+=10;return z*valf(x);}
case '+':{i+=10;return z+valf(x);}
case '-':{i+=10;return z-valf(x);}
case '/':{i+=10;return z/valf(x);}
case '\\0':{i=10;return z;}
default :{return -1000;}
}}

```

```

if((fun[i]=='a') && (fun[i+1]=='b') &&
    (fun[i+2]=='s') && (fun[i+3]=='(')
    ){i+=4;z=abs(valf(x));
switch(fun[i+1]){
    case '*':{i+=2;return z*valf(x);}
    case '+':{i+=2;return z+valf(x);}
    case '-':{i+=2;return z-valf(x);}
    case '/':{i+=2;return z/valf(x);}
    case ')':{i+=1;return z;}
    case '\\0':{i=0;return z;}
    default :{return -1000;}
}}

    if((fun[i]=='a') && (fun[i+1]=='s') &&
    (fun[i+2]=='i') && (fun[i+3]=='n') && (fun[i+4]=='(')
    ){i+=5;z=asin(valf(x));
switch(fun[i+1]){
    case '*':{i+=2;return z*valf(x);}
    case '+':{i+=2;return z+valf(x);}
    case '-':{i+=2;return z-valf(x);}
    case '/':{i+=2;return z/valf(x);}
    case ')':{i+=1;return z;}
    case '\\0':{i=0;return z;}
    default :{return -1000;}
}}

if((fun[i]=='a') && (fun[i+1]=='c') &&
    (fun[i+2]=='o') && (fun[i+3]=='s') && (fun[i+4]=='(')
    ){i+=5;z=acos(valf(x));
switch(fun[i+1]){
    case '*':{i+=2;return z*valf(x);}
    case '+':{i+=2;return z+valf(x);}
    case '-':{i+=2;return z-valf(x);}
    case '/':{i+=2;return z/valf(x);}
    case ')':{i+=1;return z;}
    case '\\0':{i=0;return z;}
    default :{return -1000;}
}}

if((fun[i]=='a') && (fun[i+1]=='t') &&
    (fun[i+2]=='a') && (fun[i+3]=='n') && (fun[i+4]=='(')
    ){i+=5;z=atan(valf(x));
switch(fun[i+1]){
    case '*':{i+=2;return z*valf(x);}
    case '+':{i+=2;return z+valf(x);}
    case '-':{i+=2;return z-valf(x);}
    case '/':{i+=2;return z/valf(x);}
    case ')':{i+=1;return z;}
    case '\\0':{i=0;return z;}
    default :{return -1000;}
}}

return -1000;

```

```
}
```

```
int main(){  
    cout<<"ENTER FUNCTION"<<endl;  
    gets(fun);  
    double x,y;  
    while(true){  
        cout<<"ENTER X:"<<endl;  
        cin>>x;  
        X=x;  
        y=valf(X);  
        cout<<y;  
    }  
    return 0;  
}
```