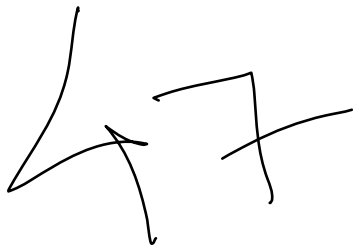
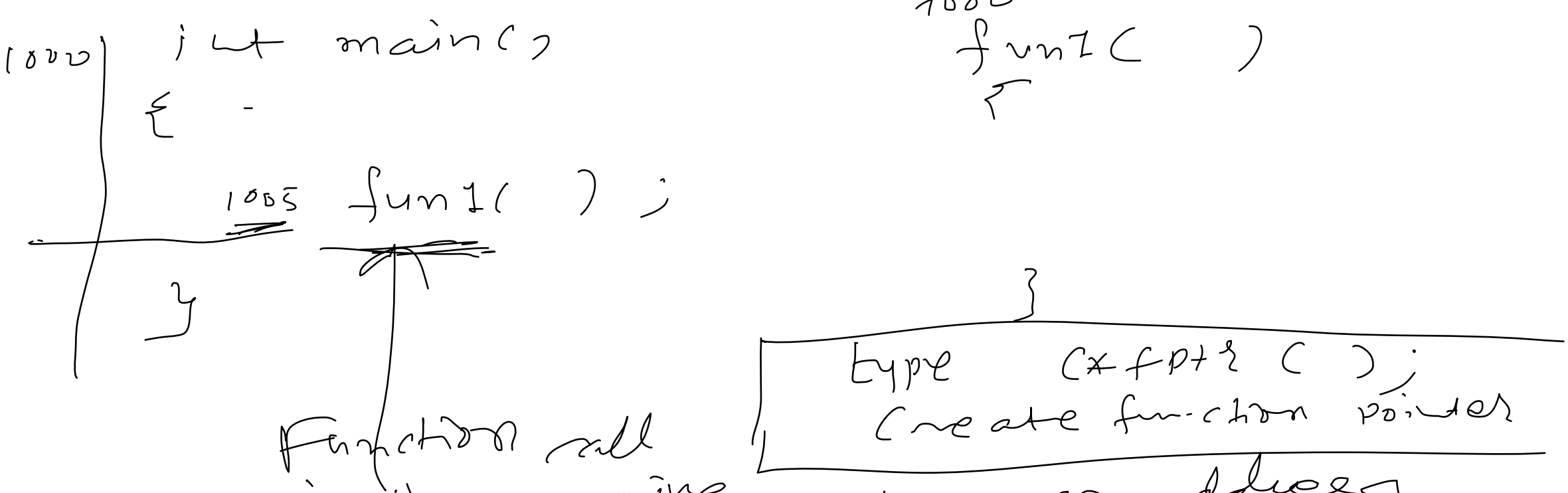
Let’s solve it



Pointer to function

Each function's binary code has some memory where it is loaded and hence the beginning address can be stored into another pointer variable.

Such pointer variable are generally called pointer to function.



Function call is like saying goto 7000 address and start executing.

Type (*fptr)( );
Create function pointer

Function name internally is itself address of 1st instruction.

fptr

address of some binary code

```
void display(int x)
{
    printf("%d", x);
}
```

```
int n=7;
display(n);
display(5);
fptr = display;
(*fptr)(n);
```

```
void dosomething
        (int y)
{
    y++;
    printf("%d", y);
}
(*fptr)(5);
fptr = dosomething;
(*fptr)(n); // dosome(n);
```

(*fptr)(n); // display

domain

⇒ double multiplication(int, int); // Read fn prototype

returntype is of importance

⇒ double (*p1)();

p1 = multiplication;

I have not specified arguments.

int x=10;

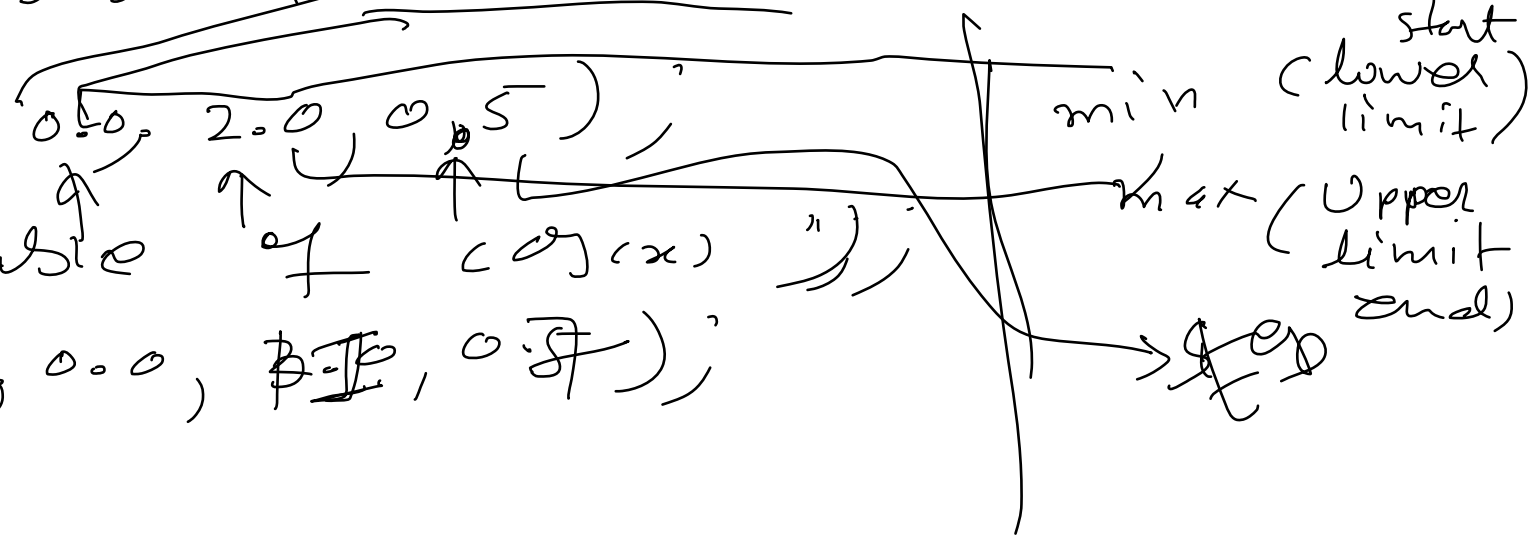
int *ptr=&x;

int *ptr2;

ptr2=ptr1;

```
double y(double);
double cos(double);
double table(double (*f)(), double, double, double);

int main()
{
    printf("table of y(x) = 2x^2 - x + 4");
    table(y, 0.0, 2.0, 0.5);
    printf("Table of cos(x)");
    table(cos, 0.0, 3.14, 0.5);
}
```



Research

Library function

exp1 exp2 exp3 exp4

cos sin

multithreading

thread library function uses function pointers.

If the ~~fa~~ar actual argument to fA is a function name fB, what another should be the datatype of this argument.

Function call

fA (fB , 10 , 20.9);

Prototype of

fA (int (*xamy) () , int , float)

int fB (int x , char y);

AJAX
tool
callback function.

```
double f( double x)
{
    return( 2* x * x - x + 1);
}
```