

Function
 ↳ Declaration → before main Function
 ↳ Implementation → after main Function
 ↳ Call of Function → inside main Function

int add(int x, int y);

```
int add(void);
int main()
{
  int x = add(1, 2);
}

int add(void)
{
  int x = 10, y = 20;
  int z = x + y;
  return z;
}
```

int add(int x, int y) ←
 10 20
 5 6
 int z = x + y;
 return (z);
 30

```
int main()
{
  int b = 10, c = 20;
  int x = add(5, 6);
  → add(10, 20);
  → add(11, 12);
}
```

→ Error

Function
 ↳ files
 ↳ bubble
 ↳ binary
 Task - binary for
 number of nodes
 Result
 run - 91
 pointer

Function
 ↳ Declaration → before (main) function
 ↳ Implementation → after main function
 ↳ Call of function → inside main function

Entry Point

```

Lab1.c
#include <stdio.h>
#include "Operation.h"
int main()
{
    ADD(5,6);
    SUB(5,5);
}
  
```

```

Operation.h
void ADD(int x, int y);
void SUB(int x, int y);
  
```

```

Operation.c
#include <stdio.h>
void ADD(int x, int y)
{
    int z = x + y;
    printf("%d", z);
}

void SUB(int x, int y)
{
    int z = x - y;
    printf("%d", z);
}
  
```



```

int main()
{
    int x=3;
    int kareem[5] = { 60, 70, 80, 1, 5 };

```

```

    for (int i=0; i<size; i++)
    {

```

```

        printf("%d \n", kareem[i]);
        printf("%p \n", kareem[i]);
        printf("%p", kareem);
        printf("%p", kareem[i]);
    }

```

Address location

0x100	0	60
0x200	1	70
0x300	2	80
0x400	3	1
0x500	4	5

```

60  0x100  0x100  0x100
70  0x200
80  0x300
1   0x400
5   0x500

```



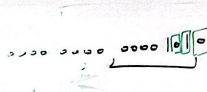
```

int main()
{
    int x = 0;

    for(char i = 0; i < 32; i++)
    {
        int z = Read_bit(
            (X) (i));
        printf("%d\n", z);
    }
}

```

0
1
0
1
0
0
0



char Read_bit(int Num, char bit_num)

Char X = (Num) bit_num

return X

2 1

Function
files
bubble
binary
Task - binary
a tree
Real
m - 9
Pentel