

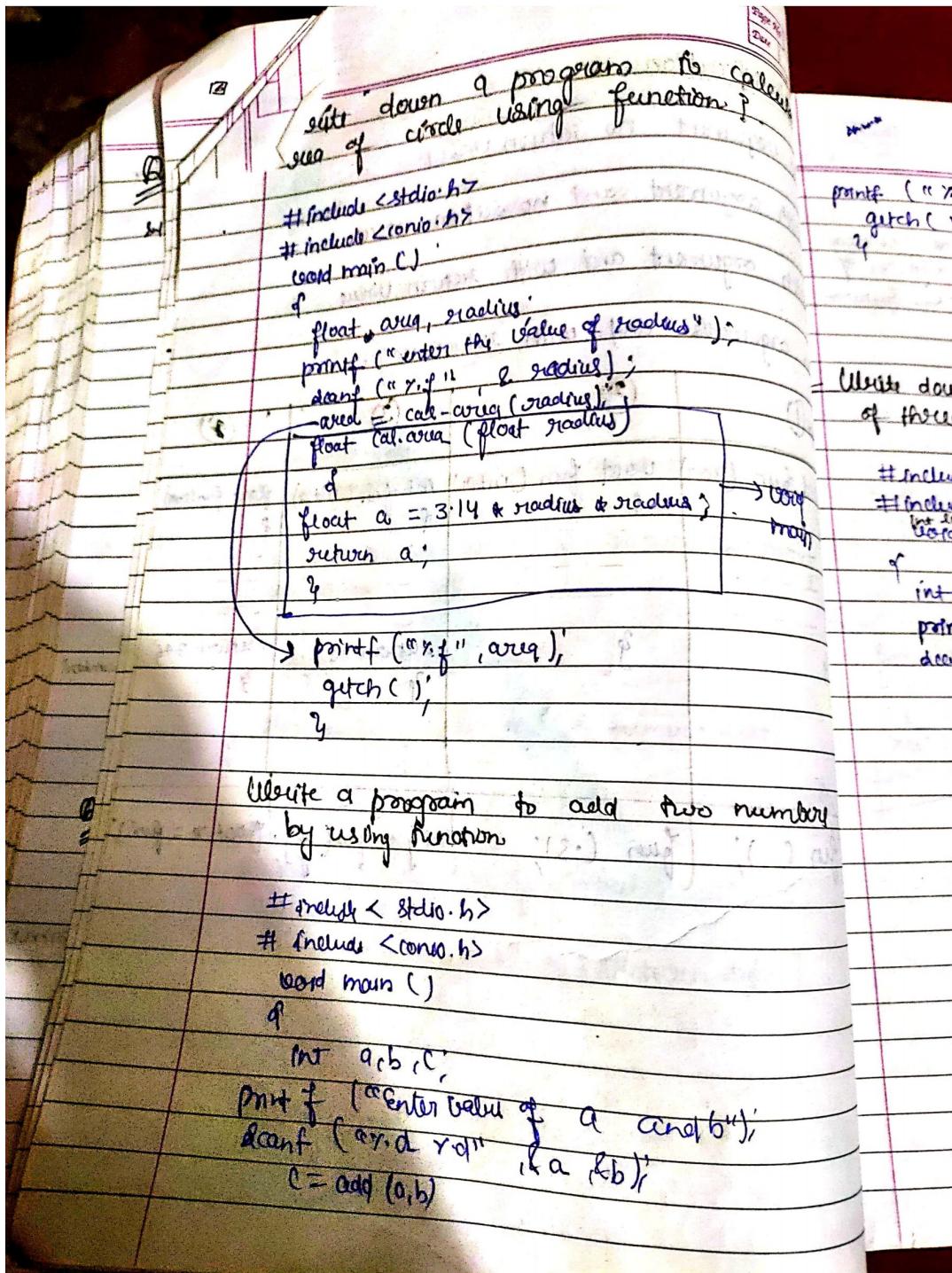
Classification

- ① No argument no return value
 - ② With argument and no return value
 - ③ With argument and with return value
 - ④ No arguments and with return value

~~Pro rango~~ (1) (2) (3) (4)

word fun (word)	word fun (int*)	int fun (int*)	float fun (void)
f	q	q	q
—	—	—	—
—	—	—	—
—	—	—	—
4	3	return 13;	return 3.4;
4	3	4	3

fun (); (fun (·3); int y = fun(y), yout n = fun(

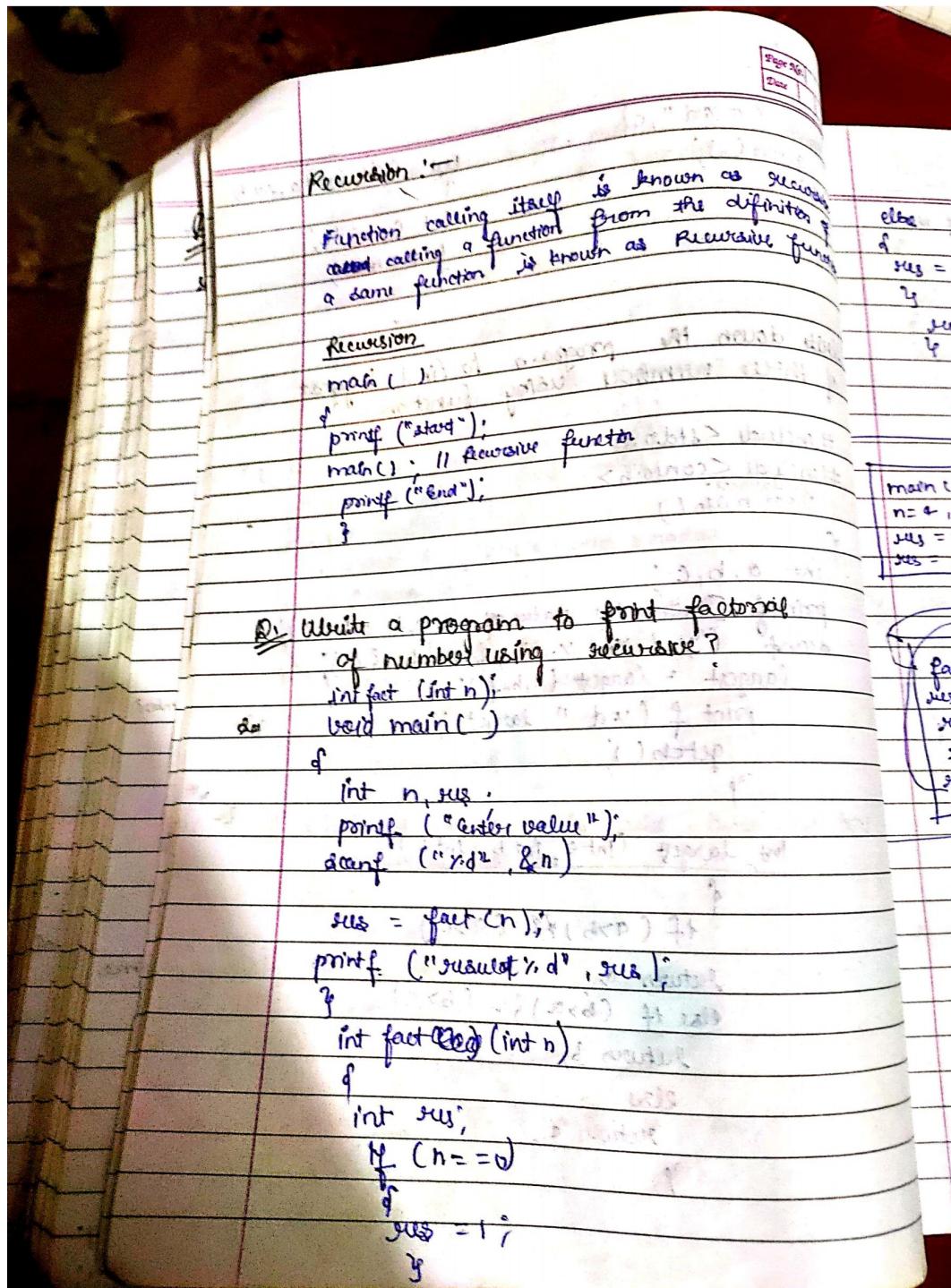


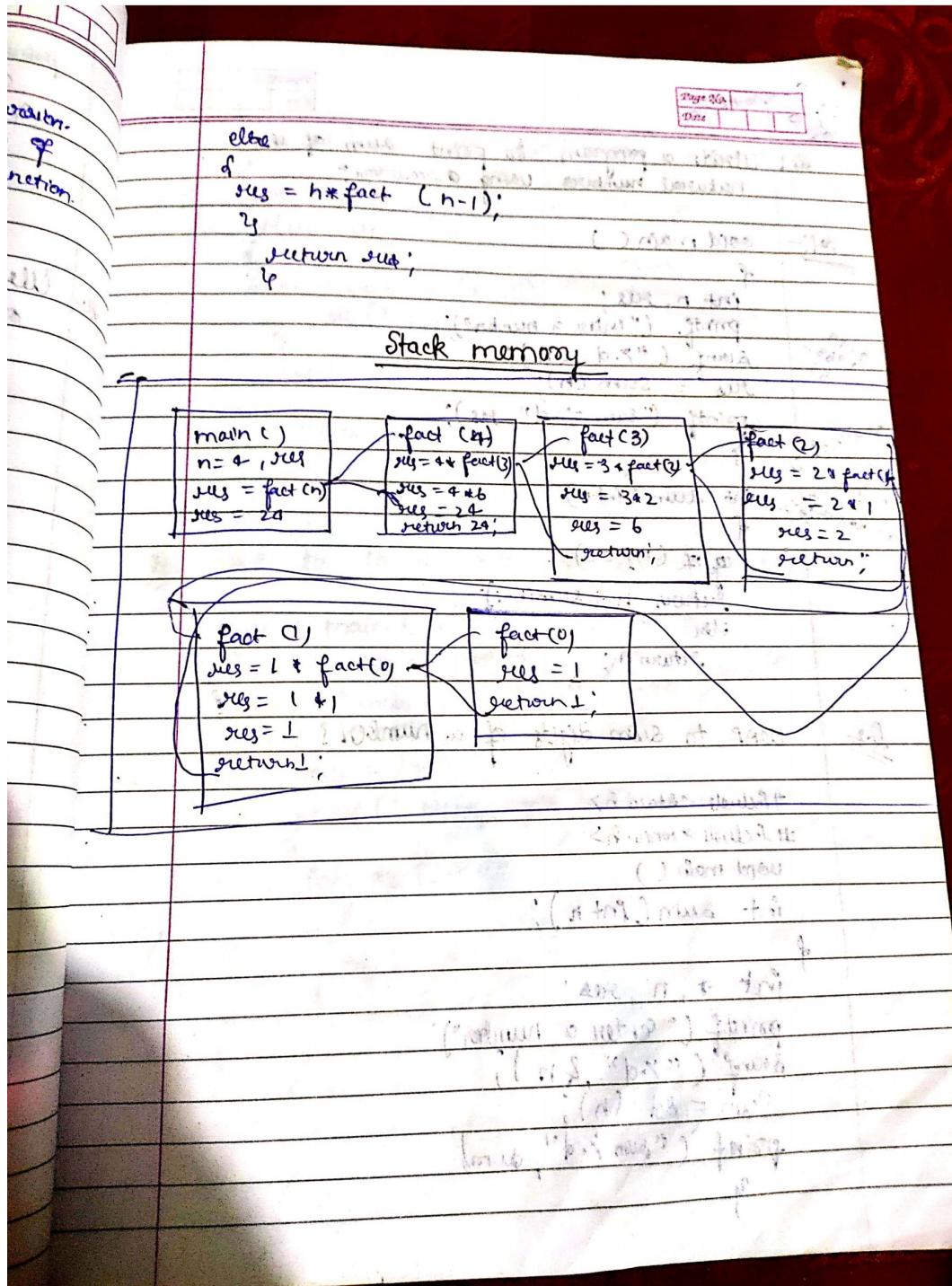
```
printf (" %d ", c)
getch ();
}
int add (int a, int b)
{
    int c = a + b;
    return c;
}
```

(ii) Write down the program to find biggest of three numbers using function.

```
#include <stdio.h>
#include <conio.h>
int largest
void main ()
{
    int a, b, c;
    printf ("Enter the value of a, b and c");
    scanf ("%d %d %d", &a, &b, &c);
    largest = largest (a, b, c);
    printf ("%d", largest);
    getch ();
}

int largest (int a, int b, int c)
{
    if (a > b) && (a > c)
        return a;
    else if (b > a) && (b > c)
        return b;
    else
        return c;
}
```





Q1 Write a program to print sum of the natural numbers using recursion?

```
def:- void main()
{
    int n, res;
    printf ("enter a number");
    scanf ("%d", &n);
    res = sum(n);
    printf ("sum=%d", res);
}

int sum (int n)
{
    if (n == 0)
        return n + sum(n-1);
    else
        return n;
}
```

Q2 WAP to sum digits of a number?

```
#include <stdio.h>
#include <conio.h>
void main()
{
    int sum (int n);

    int r, n, res;
    printf ("enter a number");
    scanf ("%d", &n);
    sum = dd (n);
    printf ("sum %d", sum);
}
```

```

1' int sd (n)
{
    int sum = 0;
    if (n > 0)
        {
            sum = sum + (n % 10);
            sd (n / 10);
        }
    else
        {
            return sum;
        }
}

```

Q WAP to count digits of entered number?

```
void main () {  
    int n, digits, count = 0;  
    printf ("Enter a number: ");  
    scanf ("%d", &n);  
    digits = dg (n);  
    printf ("No. of digits %d", digits);  
}
```

int dg (int n)

~~4. (n>0)~~

Q.1

```
int count digit (int num)
{
    if (num > 0)
        count++;
    count digit (n/10);
    else
        return count;
}

int count = 0;
main()
{
    int number;
    printf ("Enter number");
    scanf ("%d", & number);
    count = count digit (number);
    printf ("%d", number);
}
```

Q.2

Write a program to print fibonacci series using recursion function.

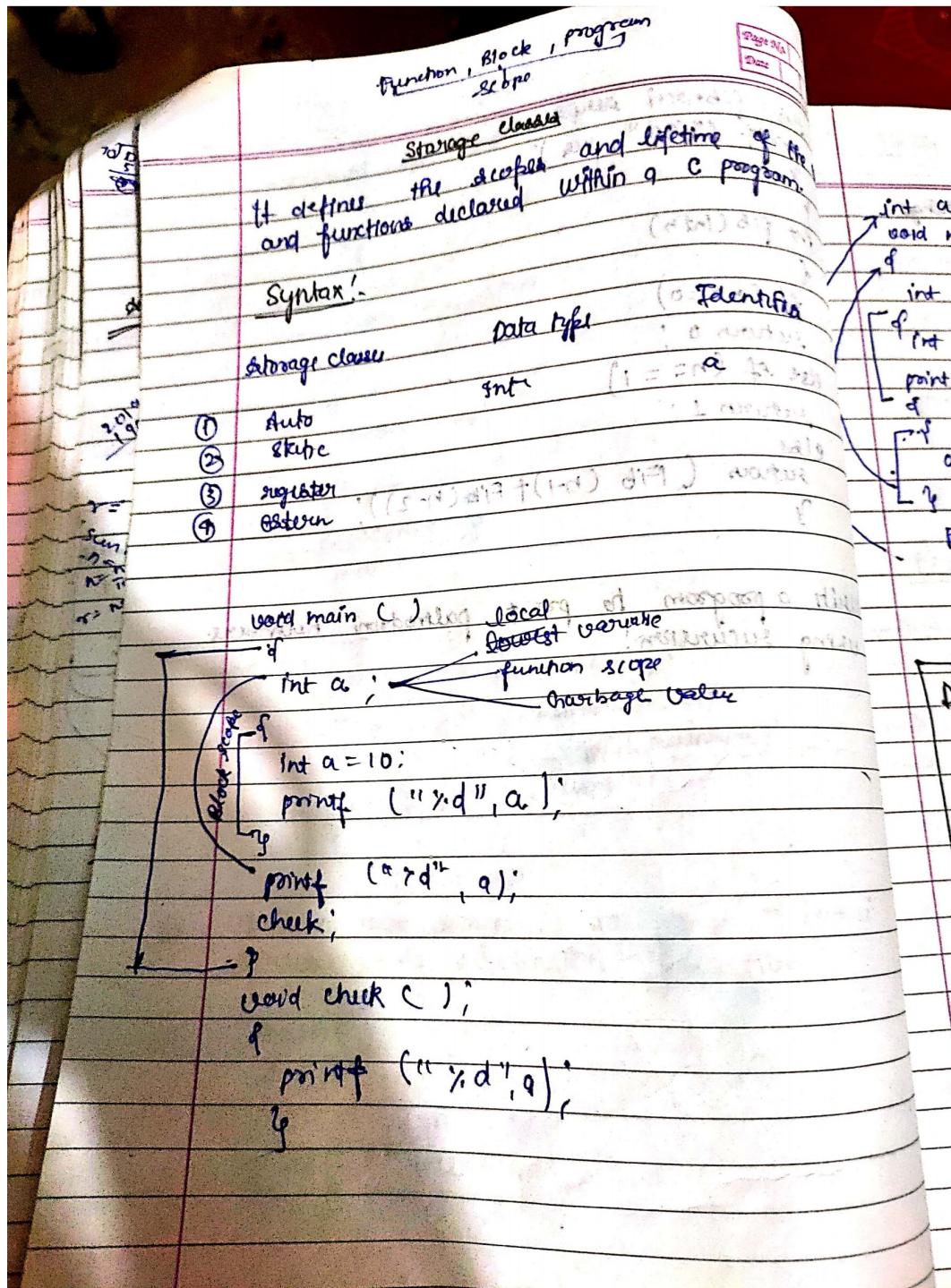
Sol:

```
main()
{
    int n, i=0, res;
    printf ("Enter no");
    scanf ("%d", & n);
    for (i=0, i<n, i++)
    {
        res = fib(n);
        printf ("%d", res);
    }
}
```

Date

ans = fibocci series

```
int fib(int n)
{
    if (n == 0)
        return 0;
    else if (n == 1)
        return 1;
    else
        return (fib(n-1) + fib(n-2));
}
```



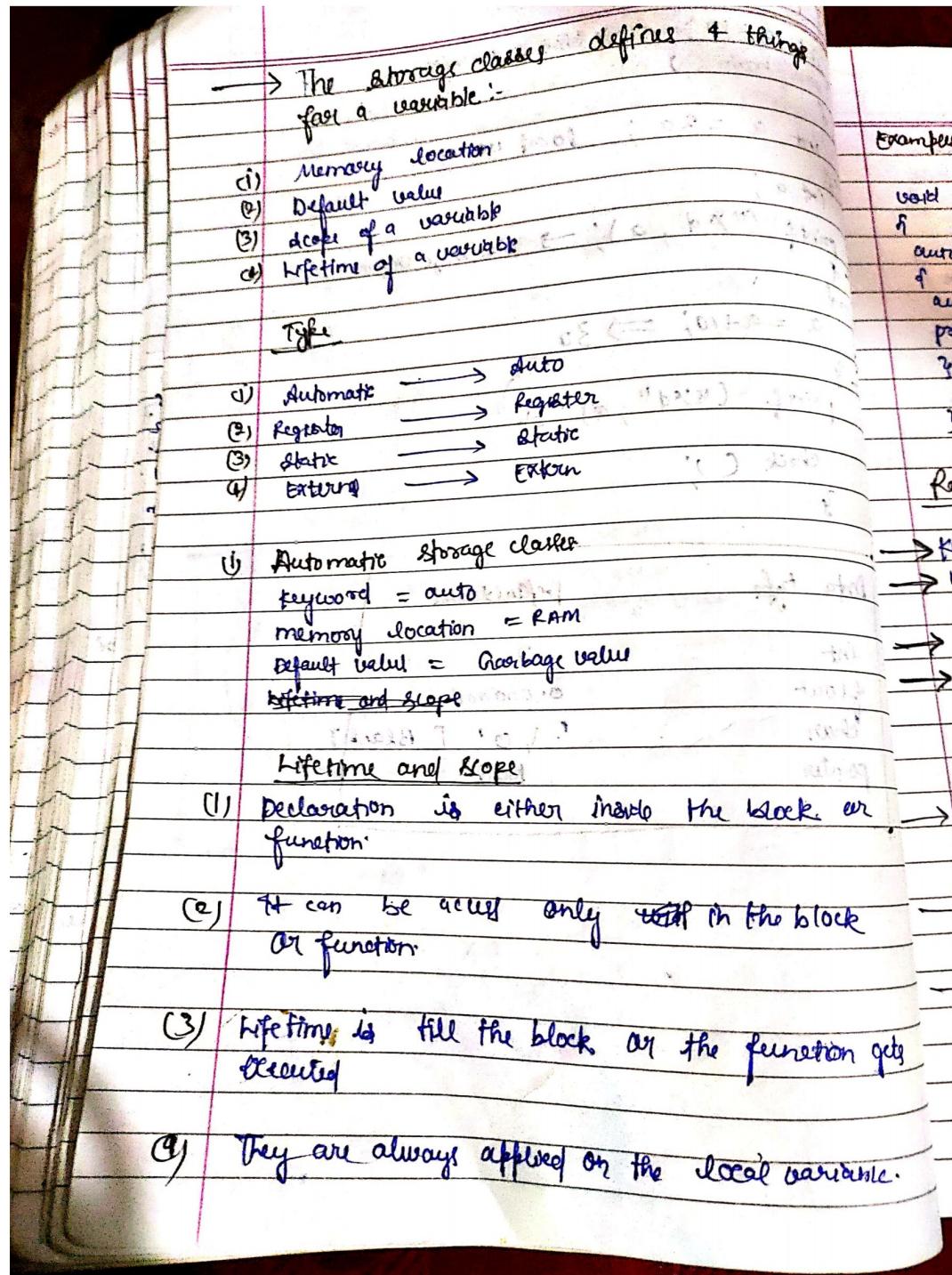
Page 56

```

int a = 10; → Global variable
void main () {
    int a = 20; local variable
    {
        int a;
        printf ("%d", a); → Garbage value.
    }
    a = a + 10; ⇒ 30
    printf ("%d", a);
}
check C;

```

Data type	Default value
int	0
float	0.000000
char	' \0' [Blank]
pointer	Null



Example:-

```
word main ()  
{  
    auto int a=10;  
    {  
        cout << "Value of a is " << a;  
        cout << endl;  
    }  
    auto int a=10;  
    {  
        cout << "Value of a is " << a;  
        cout << endl;  
    }  
    cout << "Value of a is " << a;  
    cout << endl;  
}
```

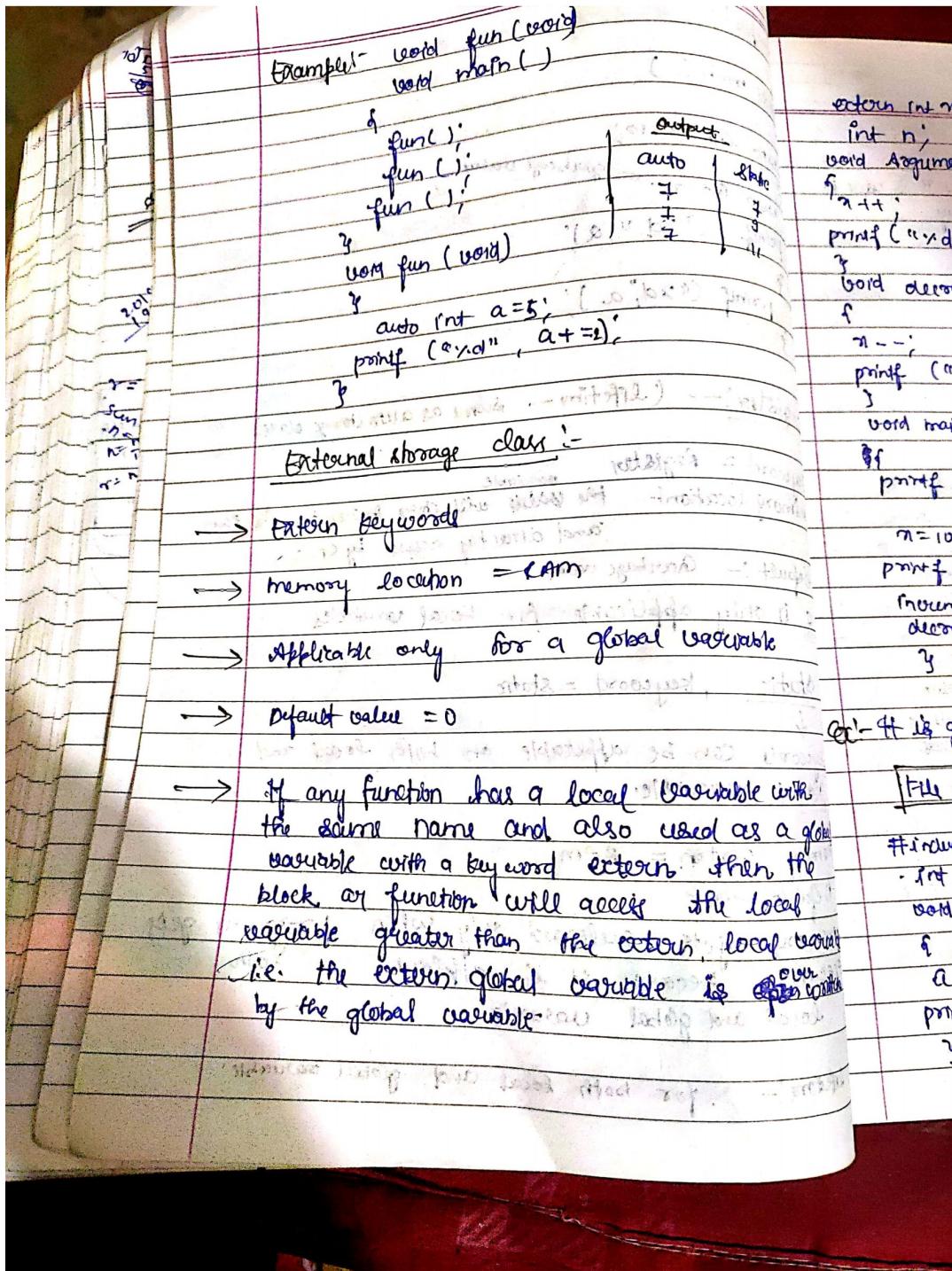
Registers:- (Lifetime — same as auto storage class)

- keyword = Register
- memory location = ^{variable} RAM
- Default :- Garbage value's = ^{variable} RAM
- It is only applicable for Local variables

static , keyword = static

- keywords can be applicable on both local and global variable.
- memory location = RAM
- Default value :- ^{variable} RAM
- memory gets released only when program gets executed because it is applicable on both local and global variable.

Lifetime:- for both local and global variable



Page No. _____
Date _____

```

extern int n;
int n;
void Argument ()
{
    n++;
    printf ("%d", n); - 11
}
void Decrement ()
{
    n--;
    printf ("%d", n); - 10
}
void main()
{
    printf ("%d", n); → 0 - default
}

n = 10;
printf ("%d", n); → 10
increment ();
decrement ();
}

```

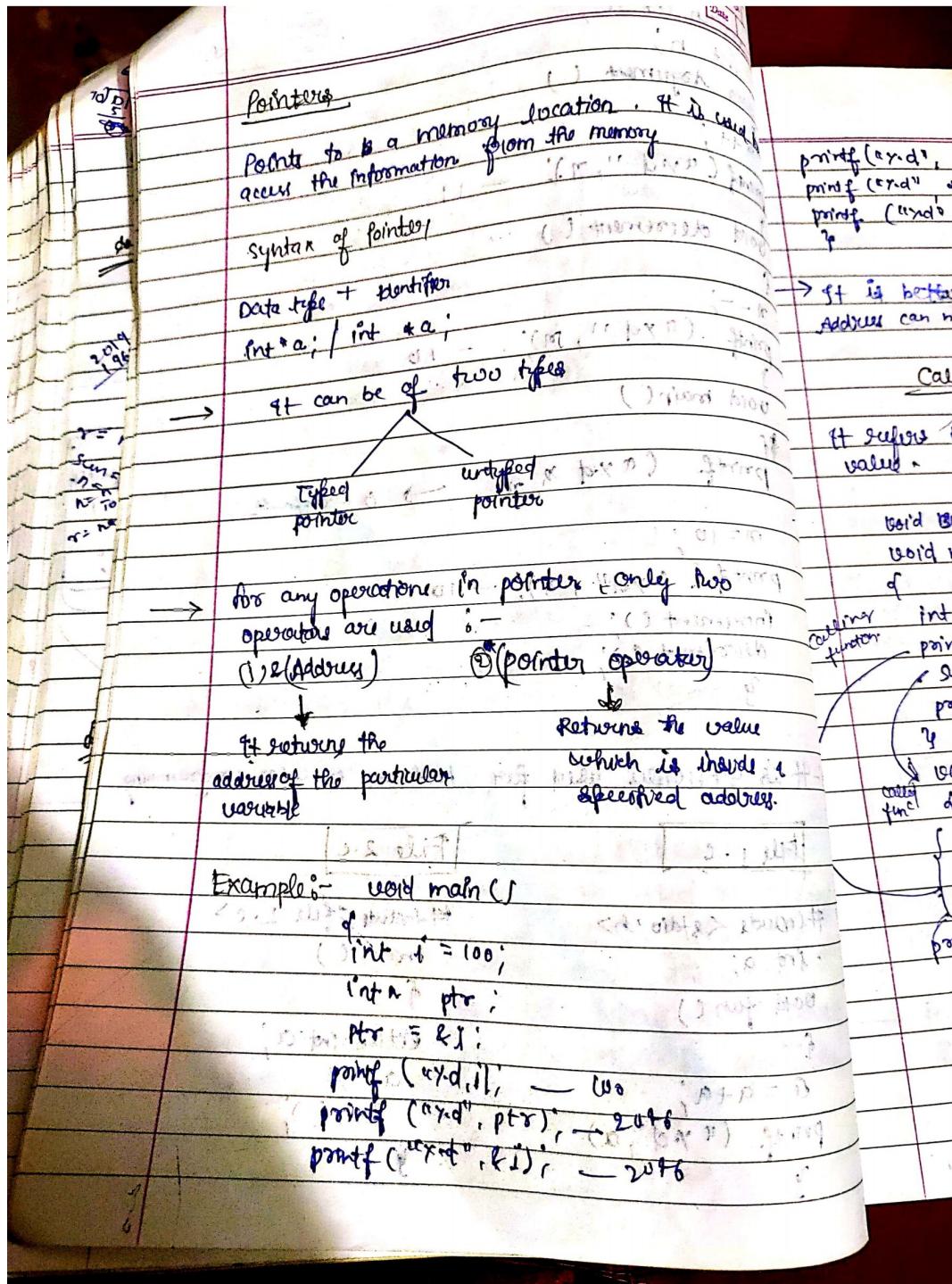
Q:- It is generally used for ~~modular~~ modular programming.

File 1.c	File 2.c
-----------------	-----------------

```

#include <stdio.h>           #include <file 1.c>
int a;                      int a;
void fun()                   {
    a = a + 1;               extern int a;
    printf ("%d", a);        a = 7
}
fun();
}

```



printf ("%d", *ptr); → 3002
printf ("%d", *ptr); → 100
printf ("%d", *(fr)); → 100
%

→ It is better to use unsigned int for address.
Address can not be negative value.

Call by Value

It refers to the calling a function by passing a value.

void swap (int x, int y),

void main ()

{

 int a = 10, b = 20;
 printf ("Before swapping %d %d", a, b);
 swap (a, b);
 printf ("After swapping %d %d", a, b);

}

void swap (int x, int y)

{

 int temp = x;

 x = y;

 y = temp;

 printf ("After swap %d %d", x, y);

}

Page No. _____ Date _____

Call by reference :-

word swap (int a*, int b*);

word main ()

```
int a=10, b=20;
printf ("Before swap val of a & b", a, b);
swap (a, b);
printf ("After swap val of a & b", a, b);
```

word swap (int *n, int *y);

```
int temp = *n;
*n = *y;
*y = temp;
printf ("After swap val of x & y", *n, *y);
```

(function passing address of variable)

(function passing address of variable)

P = B

ARRAY

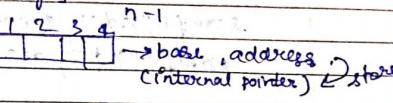
→ It stores multiple values like characters, integers, datatype float which are homogeneous.

Syntax:-

Data type Identifier [size]

int a[5]

size



→ It is the derived data type using the single array variable one can store multiple values but of same data type.

→ The name of the array is always defined as the internal pointer variable and it stores the base, address of the array.

→ int arr[5];
int arr[5] = {10, 20, 30, 40, 50};
int arr[5] = {10, 20, 30, 40, 50};



→ Global
int arr[5]; [0 0 0 0 0]

→ int arr[5];
printf("Enter an array");
for (i=0, i<5, i++)
{
 scanf("%d", &arr[i]);
}

Q1. Create a program to read and display 'n' numbers using 'n' array?

```
Q1. main()
int arr[10];
printf("Enter value of array");
for (i=0; i<n; i++)
    scanf("%d", &arr[i]);
printf("%d", arr[i]);
```

Q2. Write a program to find the sum of array elements?

```
Q2. int arr[5] = {10, 20, 30, 40, 50};
int sum = 0;
for (i=0, i<5, i++)
    sum = sum + arr[i];
printf("%d", sum);
```

Q3. Write down
of a array
length = 5

```
int b = a[0];
for (i=
```

```
i
if (
    b
)
pr
```

Q4. Write a
And:
#include
word
if
int

```
for
for
for
for
```

Q. Write down the program to find largest element of an array?
Ans: range = arr[i:j]
b = a[0]
for (i=0; i<n; i++)
if (b < a[i])
b = a[i];
printf("%d", b);

Q. Write down the program to find average of numbers?
Ans:
#include <stdio.h>
#include <conio.h>
void main()
{
int a[100], i, n;
float avg, sum = 0; // float is used for decimal
scanf("%d %d", &n);
for (i=0; i<n; i++)
{
average = sum + avg[a[i]]
scanf("%d", &a[i]);
}
for (i=0; i<n; i++)
sum += a[i];
}
printf("%f", sum/n);
getch();