SPL - Final

Final

1. Function
2. Recursion
3. String
4. Pointer
5. Structure
6. File (Read/write)

1. Function

```
int main() {
    int a,b;
    scanf("%d %d", &a, &b);
    sum(a,b)
        ↳ actual parameter
    float c;
    sum(a,c)

    sum(int x, int y);
```

Formal Parameter

```
int sum(int x, int y) {
    int res = x+y;
    return res;
}
```

* ~~get g~~
* getchar() → return char
  - 1 read inp
  - 2 then return
  - → like as scanf

* maintain the sequence

```
void example_fuction() {
    for(i=1; i<=5; i++) {
        printf("Hi\n");
    }
int main() {
    example_fuction();
    printf("end");
}
```

Hi
Hi
Hi
Hi
Hi
end

It's better to declare prototype at top of the program

```
void f1();
"   f2();   } Prototype
"   f3();
```

```
void f1() {
    f2();
    printf("f1");
}
void f2() {
    f3();
    printf("f2");
}
void f3() {
    for(int i=1; i<=3; i++) {
        printf("%d\n", i);
    }
}
int main() {
    f1();
    printf("end");
}
```

implementation

output

1
2
3
f_2
f_1
end

f1() → call / revoke

```
func(3)        func(2)      f(1)
f(2) x=3       f(1) x=2     f(0) x=1
     P(x)           P(x)         P(x)
     3
     2
     1
     1
     2
     3
```

C-14

Function (temporary something)

- → WAP (write a program)
- → Output

Local Variable

↓

```
#include<stdio.h>
int f1 ( int a) {      → formal argument
    int a,
    for (i: .... ) {
        //statment
    }
```

Global variable

↓

```
#include <iostream>
int a,
int b
void U1 ( ) {

}
int main ( ) {

}
```

---

void f1 (int a) {

a++ ;

```
}
int main ( ) {
    int a =10;
    f1 (a);
    a=50;
```

[function থেকে বের হলে track/m হারিয়ে ফেলে। তখন new value / new function এর নিয়ে কাজ হয়]

→ Qn:

```
int sum (int a, int b) {
    int result =0;
    result = a+b;
    return result;
}
int main( ) {
    printf("%d", sum (10,12));
    "      (        sum (20,25));
    "      (        sum (30,25));
    return 0;
}
```

Qn Triangle Calculation →

```
calculateTriangularNum( int n) {
// code statment


}
int main() {
    for(int i=0 ; i<3 ; i++) {
      cin >> n ;
     calaculateTriangular Num( n)
    }
```

Slide → Example (13 page)

owpwt → 9.000000

10.000000

Example (20 P)

```
| include<stdio.h>
```

owpwt → 150 & 35 = 5

1026 & 405 = 27

83 & 240 = 1

Example (global variable)

x = 9

যদি global and local variable same same থাকে, main এ আবার আগে main এর value দিবে priority পাবে আর main এ দিবে / local দিবে print হবে

⇒ Qn

code

global

a = 0 ~~6 12~~ 24

b = 0 24

c = 0 24

Output

12 0 0

12 24 24

24 24 24

func3(int c)

c = ~~6~~ 2

a = a * 2

= 12

c * a

func1(int P)

P = 12

c = P + a

= 12 + 12

= 24

x = 12, b = 24

x = 24

func3(int c)

c = ~~24~~ 2

4 mark Question from [global + Local] ***

swap

19-11-2023 C-14

String → annay of chanaeters

- → inpwt/owtpwt (%c)
- → size, length
- → "\0" → use -করলে যোগে স্বয়ংক্রিয় End

count the number of character in string.

```
#include<iostream>
int stringLength(char str[]){
    int count = 0;
  while(str[count] != '\0')
    count++;
   return count;
}
int main(){
   char word1[] = {'a','e','i','o','u','\0'};
   char word2[] = {'o','k','\0'};
   char word3[] = {'h','a','\0'};
  cout ....
}
```

declaration

```
char ar[] = {'a','b','c'... '\0'}
char ar[] = "Hellow" "world";
char ar[] = " "
char ar[], = {"abcd\
              ABCD"}
```

Character → 'A';
String → "lala";
tab → '\t ---- (4sp)

string inp/out

- → forloop
- → scanf("%20s", &ar)
- → printf("%s", ar
- → gets(ar)
- → fgets(ar+file)

string Concatanation (manually)

concate Function (manually)

```
void concat( char result[], char str1[], char str2[])

{ int i, j;
  for(i=0; str1[i]!='\0'; i++) {
     result[i] = str1[i];
  }
  for(j=0; str2[j]!='\0'; j++) {
     result[i+j] = str2[j];
  }
  result[i+j] = '\0';
}

int main() {
     char s1[] = "Test"
     char s2[] = "work.";
     char s3[20];
   concat(s3, s2, s1); → called the fuction
  printf("%s\n", s3);
  return 0;
}
```

String equality

```
bool equalString( char s1[], char s2[])
 { int i=0; bool equal;
  while(s1[i]==s2[i] && s1[i]!='\0' && s2[i]!='\0')
    i++;
  if(s1[i]=='\0' && s2[i]=='\0')
     equal = true;
  else
     equal = false;
  return equal;
}
```

String Compare

S1 < S2

S1 > S2

String Copy

Class-16 - 22-11-23

Build in Function :

① Strlen (s1) → char s1[ ] = 'hello' ;

② Strcat ((s1), s2) → s1 = s1+s2

result stored here.

destination

s1 = "Mustiqur"
s2 = " Rahman" } → s1 = Mustiqur Rahman

③ strcpy (s1, s2) → s1 = s2 → 2nd string value 1st string

s1 = Rahman ↳ copy হয়ে যাবে

s2 = Rahman

M = 77
R = 52

④ strcmp (s1, s2) s1 < s2 s1 - s2 = -ve

s1 > s2 s1 - s2 = +ve

s1 = s2 s1 - s2 = 0

char D[20]

fgets() → আগে use করা যেতো, "\n" এর আগ পর্যন্ত value নিবে

gets → কাউকে ব্যবহার করতে দেয় না ↳ Both take input including space

scanf → input নিবে up to space

(v) strrev(s1) = s1 = "world"

↳ s1 = dlrow

(vi) strncat(s1, s2, n) s1 = hellow
s2 = world
n = 3

n = number of letters to do the operation

s1 = hellowor
s1 = world

(vii) strncpy(s1, s2, n)

s = s2
n = 3
s1 = wor
s2 = world

(viii) strncmp(s1, s2, n) # atoi()

(ix) strchr(s, c)

Sn

	Str1	Str2	Str3
	Hellowond	Programming	Hello wo
	"	ProgrammingHello	programmingHell
	"	"	
	"	"	Pronominin

Q1

str1	str2	str3
Hello world	Programming	Hello wo
"	Progrmming Hell	"
"	"	Programming Hell
"	"	Programming Hell Hell
"	Pr	"

Q2

str1	str2	i	k
0 1 2 3 4 5 6 7 8 9 10 11			
HELLO FELLAS	BEST	6	-
Hello BELLAS	"	"	0 ← for loop
" [E=E]	"	6	1
str1[i+k] = str2[k] Hello BE$LAS	"	6	2
str[6 to 0] = str2[0] Hello BESTAS	"	6	3
SATSEB OLLEH	"	6	-
SATSEB OLLEHBEST	"	-	-

if(strcmp(s1,s2)>0)

B-S = -ve ✗ → SATSEB OLLEHBEST UIU IS THE BEST - -

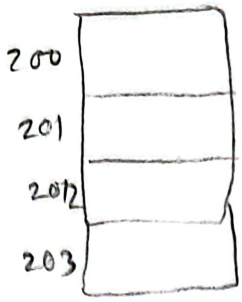
else ✓

strcpy(str2, "UIU is THE BEST");

Q. WAP

11-28-23

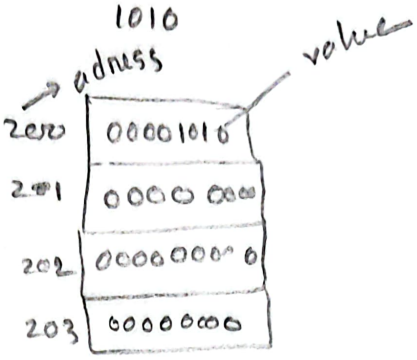
Pointer



int = 4bit,

int a = 10

1010



Pointer dec

int *p * → dereference operator

P = &a → adress

*p → vale

int *p

int a = 10

P = &a

P = 200

*p

a → value

&a → address

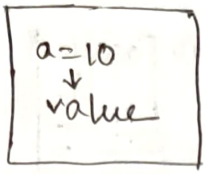
p → adress

p → address

* p → value

11-29-23

"Pointer"



4097... → address

int* p ; / int *p; → declaration at first, it called address

*p = 4560

print ("%d", p)

print ("%d", *p)

p = 500 [a = ~~10~~ 20] 700 [b = 20]

int *p ;

int a = 10;

p = &a

if p = b

int b = 20;

*p = b ;

(// a = b)

p	*p	a	b	&a	&b
↓	↓	↓	↓	↓	↓
500	20	20	20	500	700

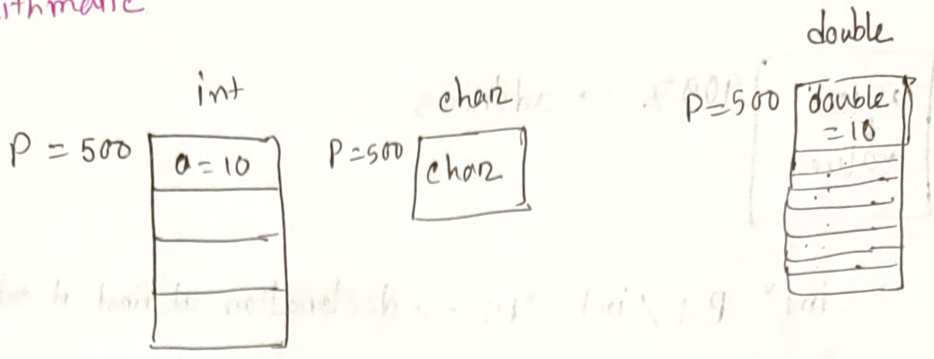
a → value

&a → address

p → address

*p → value

Pointer Anithmatic



printf("%d", p+1);
↳ 504

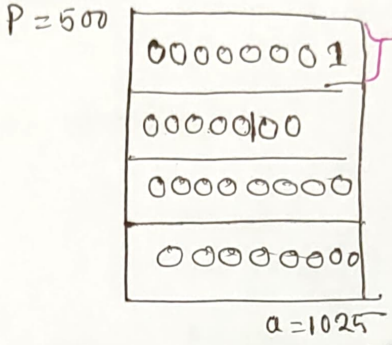
"
↳ 501

"
↳ 508

P=500 a = 10

*P+1 = //10+1 = 11

*(P+1) · // garbage
↳ 504 nuber value



annay → address pass

P0 = 500 → address

*P0 = 1 → value

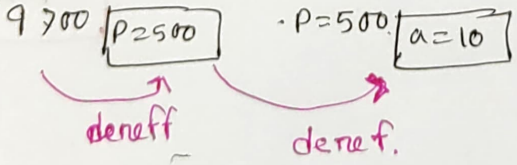
P0 + 1 → 500+1 → address
= 501

*(P0+1) → 501 → value
→ 4

Pointer to Pointer:

int **q;

int *** r



r = 900 q = 700

r // 900

*r // 700

**r // 500

***r // 10

p = &a
q = &p
r = &q
r → q → P → a
↳ processing Simulation

P // 500

*P // 10

q // 700

*q // 500

**q // 10

#

```
int x=20;
int *p;
int **a
int **b
int ***z
```

z=&a

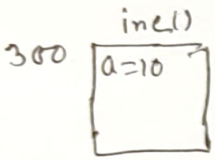
p=&x

a=&p

z → a → p → x

#. Call by value

. call by reference




```
void inc (int a)
    a = a+1
    printf("%d", a); // 11
}        "   &a  // 500(un)
int main(){
    int a=10;
    int c(a);
    print("%d\n", a); // 10
        "          &a // 300
    return 0;
}
```

↓

called by value

```
void inC (int *p) {
    *p = *p+1
    printf("%d", *p); // 11
}
int main(){
    int a=10;
    int c. (&a);
    printf("%d", a); // 11
```

call by reference

7-12-023

Pointer Array.

int an[] = {2,3,5}

ann

↓

&ann[0].

int *p = ann

= &ann[0];

A[i] ⇒ value

&A[i] → address

A+i → address → A → &A[0].

*(A+i) → value

A+1 → &A[0]+1

P+i → address

A+2

*(P+i) → value

A+3 → index

	400	404	408	412	416
A	2	3	5	7	9

A+4 → 416

*(A+2) → 5

Dynamically Allocation

→ A+1 = &A[1]

&A[0] +1

&A[0]	&A[1]	
2	3	4

malloc

realloc

calloc

(int*) (malloc())

f.

*(a+i+1) = *(a+1-i) - (n2+i)

*

output :

5

13

19

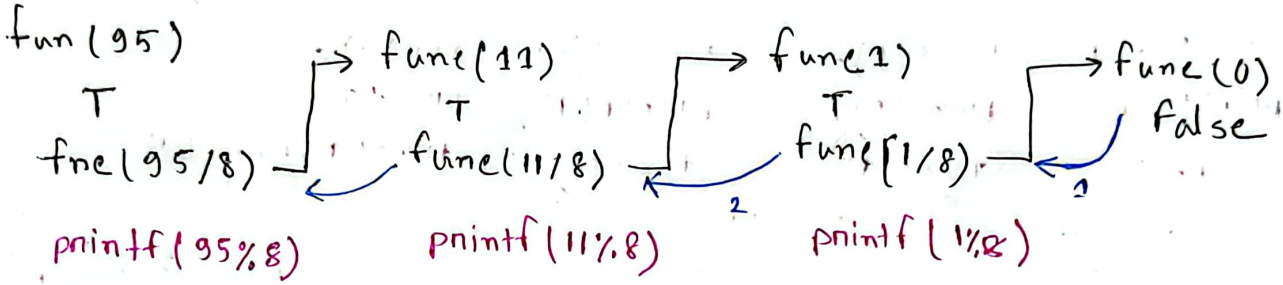
matrix manual stress

a[0]	a[1]	a[2]	a[3]	a[4]	a[5]	a[6]
6	3	2	7	0	1	5
6	3	1	-7	0	1	5
6	3	1	-7	0	1	5
6	3	1	-3	0	1	5
6	3	1	-3	0	1	-9

8/11/11

Recursion output

```
void function (int num) {
    if (num>0) {
        function (num/8);
        printf("%d", num%8);
    }
}
```



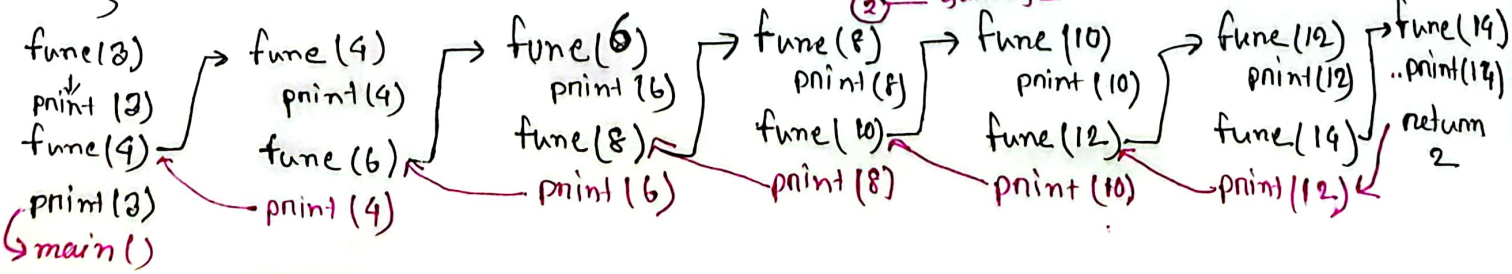

```
int fune(int n) {
    printf("%d\n", n);
    if (n%17==0)
        return 2;
    else if (n%2==0)
        fune(n+2)
    else
        fune(n+1)
    -> printf("%d\n", n)
}
void main(){
    printf("%d", fune(3));
}
```

output

137

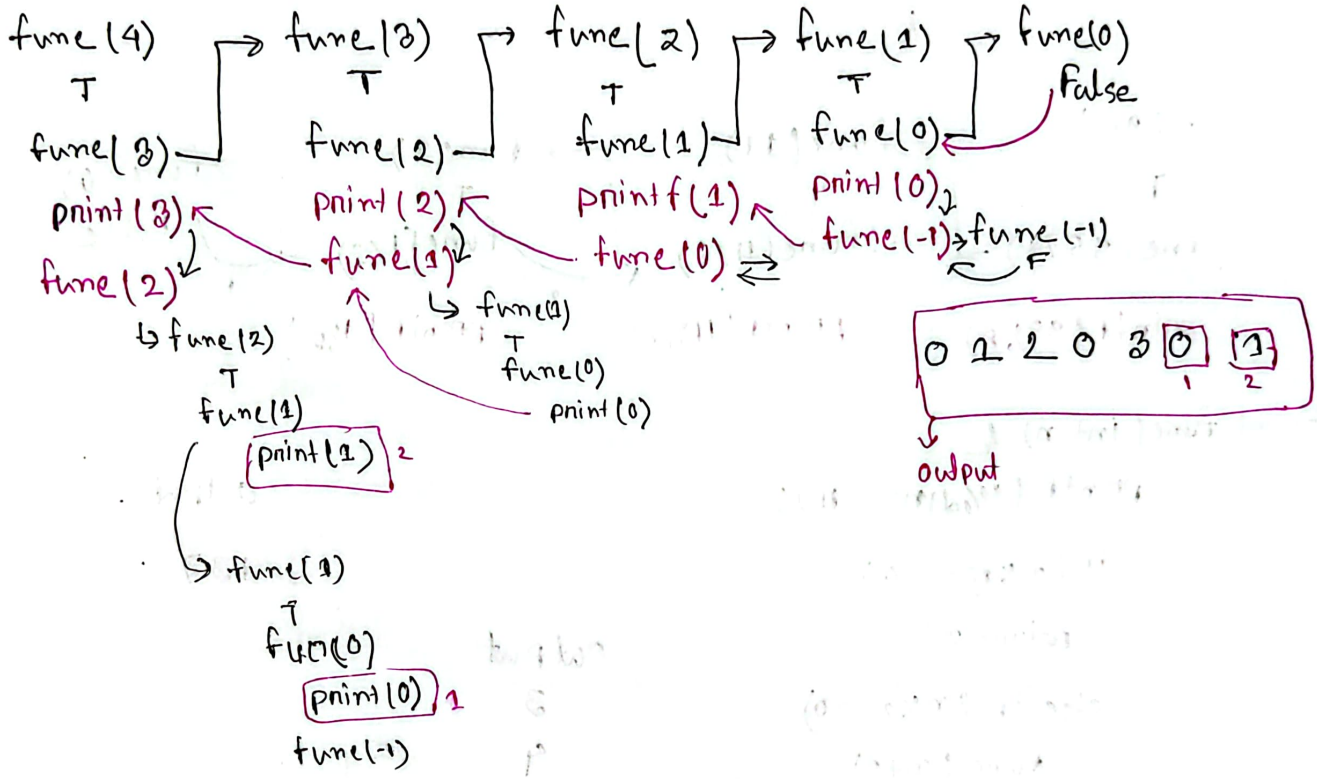
Output

3
4
6
8
10
12
14
12
10
8
6
4
3
2 — garvage



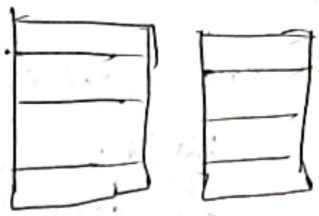
num = 4

```
# void func (int num) {
    if (num > 0)
    {
      func(--num);
      printf("%d", num);
      func(--num)
    }
}
```



main() done

Structure



```
struct student {
        char name[20];
        int id;
        float cgpa;
};
```

user inputs

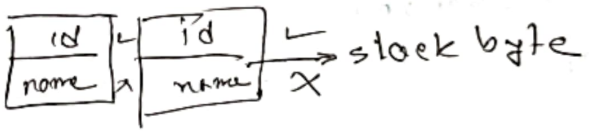
```
fgets(st1.name, 20, stdin)
```
size → 20; stdin → standard input

```
scanf("%d", &st1.id);
scanf("%f", &st1.cgpa);
```

copy →

```
s2 = s1
```

compare →

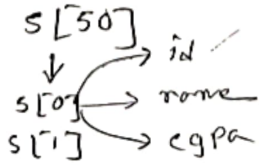


~~s1 = s2~~ → X

```
if(( st1.id == st2.id) && (st1.cgpa == st2.cgpa)
    && (strcmp(st.name, st2.name)) == 0
    print("Both are equal")
```

Array of structure

```
struct student s[50];
```



single output

```
s[4].name
```

multiple output

```
for(i=0; i<50; i++) {
    printf("%f", s[i].cgpa);
```

```
struct stdn {
    char name[50]
    int id;
    float cgpa;
    float cl[6];
};struct student S[50]
```

nested structure

```
map
↓
max
min
avg
sum
```

pointer to array

```
*(e1).name
e1→ name
```

Passing ↓

- member variable
- array

String WAP

① Change every Cogonent .

```
input
  fgets ( message , sizeof(message), stdin)

for(i=0 ; *message[i] != '\0' ; i++)
        ch = message[i]

    if (isalpha(ch)) {
        if (ch>='A' && ch<='Z')

          if (ch != 'A' ....
            if (ch == 'Z')
                ch = 'A'
            else
                ch++ ;

    message[i] = ch
```

⑪ Substring match

```
length → int length =0
          while ( str[length] != '\0") {
                length ++;
          }
          return length
    }

for(i=0 ; i<= mainlen - sublen ; i++) {
    for( j=0 ; j<sublen ; j++) {
        if ( mainString[i+j] != substring[j] )
            break;
    if ( j == sublen {
        found =1
        break;
```

File

Read

Writes 

Appends 

gets → fgets

gete → fgete

puts → fputs

pute → fpute

printf → fprintf

scanf → fscanf

fopen()

fclose()

```
int a;
int *p;
p = &a;
```

fopen → file ka pointer return karta hai

```
File *fp;
fp = fopen("D:\\student\\student.txt", " ")
```

Location /filename

operation mode

r	r+
w	w+
a	a+

```
char s[50] = "hello";
fputs(s, fp)
.
.
.
fclose(fp);
```

Write → w

```
char s[50] = "hello";
fputs(s, fp);
```

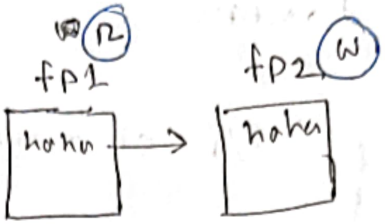
read → r

```
char s[50] = "hello";
fgets(s, 50, fp);
```

একটা file থেকে Read করে অন্যটায় write.

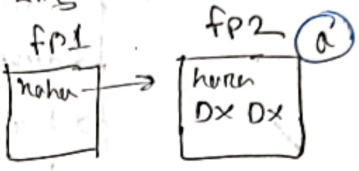
File* fp1 , *fp2

```
fp1 = fopen
fp2 = fopen
char s[50];
fgets(s, 50, fp1)
fputs(s, fp2);
fclose(fp1);
fclose(fp2);
```



*p1 2

append করলে fp1 to fp2 (শেষে যোগ হবে)

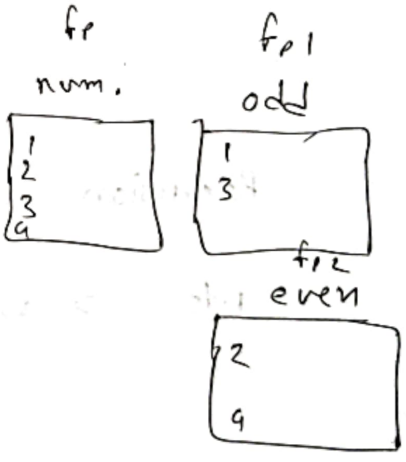


fprintf(fp, "Name = %s, ID = %d",

fscanf(fp, %d %s %f ", &id, &name, &, cgpa)

```
int
fflush(stdin);
string/char
    int n;
while(fscanf(fp, "%d", &n) != EOF) {
    if (n%2 != 0) {
        fprintf(fp1, "%d", n);
    else
        fprintf(fp2, "%d", n)
}
```

num → characters



Final

Function
- → O/P or manual tress
- → WAP

String
- → O/P or manual tress (using Library func)
  - ↳ strcmp(s1, s2) </>
- → WAP
  - ↳ vowel
  - ↳ consonent
  - ↳ Revr
  - ↳ occurance
  - ↳ toupper
  - ↳ tolower
  - ↳ sub string

a = 97 → to A = a − 32

A = 65 ↳ to a = A + 32

*str(++)

Structure → WAP
- ↳ array print
- ↳ input
- ↳ multiple input → student[i].mark[j]
- ↳ calculation (max, min) (avg, sum)
  - ↳ max, min

Recursion → output

file → WAP
- ↳ numbers
- ↳ characters

console → file → w, r

file → console →

file → file → console

a+ → কিছু file এ থাকলে তার পরে লেখবে

a+, w+ → write

w+ ↳ write (new)

(r+) → write / read [file exist না থাকলেও 0]

Pointer
- → O/P or manual tress
- ↳ wap (call by reff)
  - (array) * *

Structure

Structure Define

```
struct student {
    char name[20];
    int roll;
    float mark;
    float cgpa;
};
```

Structure User input

```
fgets(std1.name, 20, stdin);   // for string inp
scanf ( .... )
```

Accessing the member

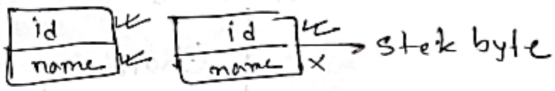
```
struct student std1;
```

Structure copy

```
student 2 = student 1
```

compare

```
S1 = S2
```




```
if ((st1.id == st2.id) && (st1.cgpa == st2.cgpa) &&
    strcmp(st1.name, st2.name)) == 0
{
    printf("Both are equal");
}
```

Array of structure

```
struct stdn : S[50]
    ↳ S[0] → id
      S[1] → name
      S[2] → cgpa
```

Pointer

```
*(S1).name
```

File

File

① File define

File *fp

(2.i) if(fp== NULL)
printf ("cant' open fille");

(ii) File open

fp = fopen (" mus.txt ", "made")   (location)

→ made are
- Read → "r", "r+" (read and write)
- write → "w", "w+" (write and read)
- appends → "a", "a+" (append and read)

//..... statment .....

(iii) File closing

fclose()

(iv) File write

char s[50] = "hi kemon acho";

→ fputs (s, 50, fp)

* After taking the input use
fflush(stdin);
→ it clear the unwanted char...

(v) File read

char s[50] = "valo na";

→ fgets (s, 50, fp)

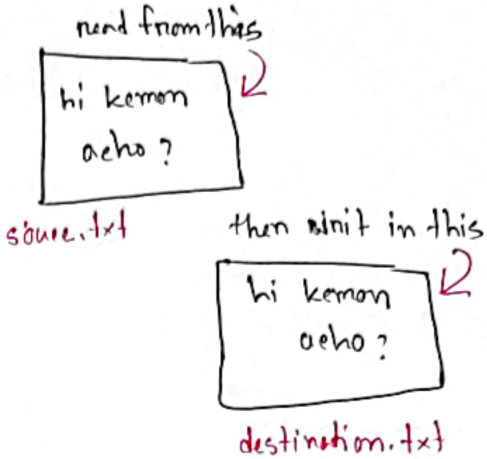
(vi) → Single charecter write and read

```
fp = fopen ("file1.txt", "w"); char c;

if ( fp == NULL) {
    printf(" Sorry ")
    exit();
}
else {
    printf ("Enter the text until Enter-key : \n");
    while((c = getchar())! = '\n')   // read → while((c=getc(fp))! = EOF) {
    fputc (c, fp)                    //           fgetc (c, fp);
}                                    //         }
fclose(fp);                          //         fclose (fp)
```

① Read one the write another.

```
FILE *srcF *destF;
  char ch;
srcF = fopen("source.txt","r");
if(srcF == NULL){
        printf("can't open");
        break;
}
destF = fopen("destination.txt","w");
    if(destF == NULL) {
            // statment
    }
while((ch = fgetc(srcF)) != EOF) {
        fputc(ch, destF);
}
fclose(srcF);
fclose(dstF);
```



* same process for other using o fgetc / fputc
  - vowels
  - consonant
  - characters

Logic in your mind 😕

* for string
  - fgets(
  - fputs(