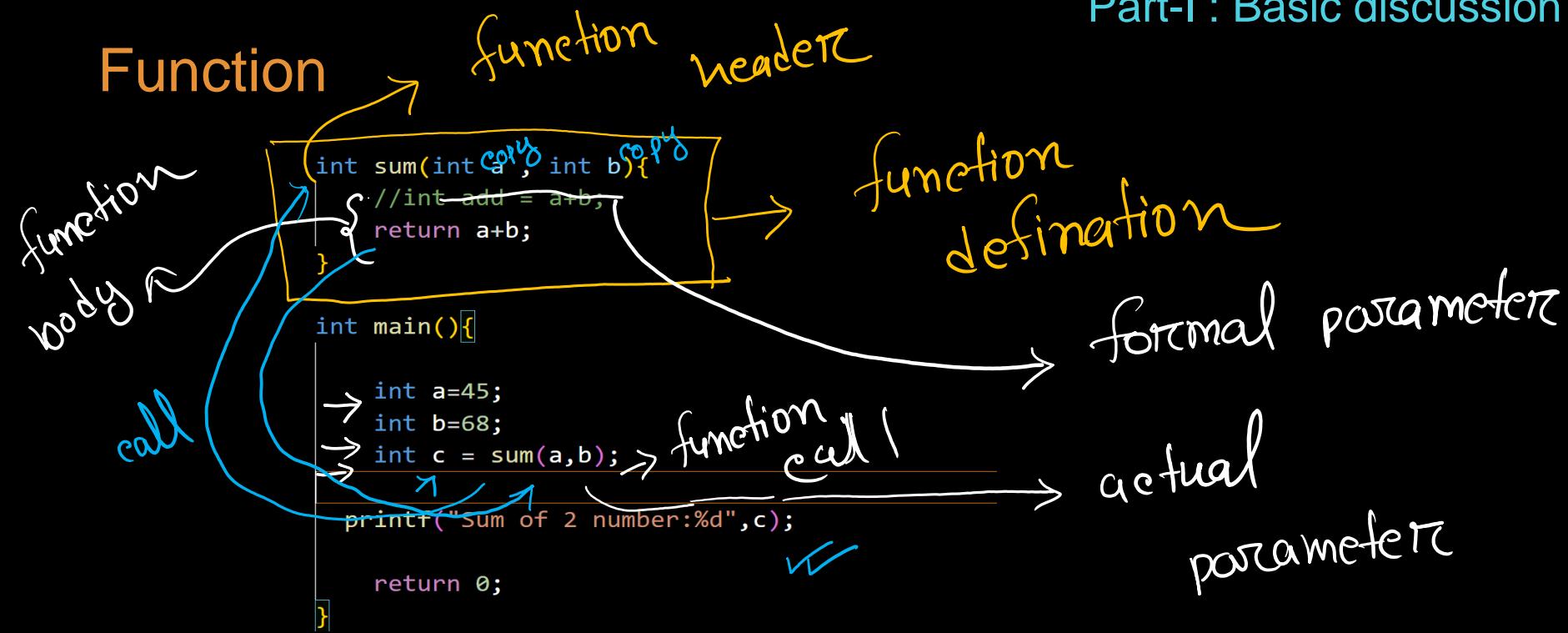


# Question pattern

1. Array
2. Pointer
3. String (char array)
4. Function
5. Structure
6. File (read, write)

} • manual  
✓ tracking  
( popular )



```
void maximum(int a → 45, int b → 68, int c → 95){  
    ✓  
    if(a>=b){  
        printf("%d",a);  
    }  
    else if(b>=c){  
        printf("%d",b);  
    }  
    else{  
        printf("%d",c);  
    }  
    ✓  
}
```

main( )

a = 45,

b = 68

c = 95

95      68      95  
            

45 > 68  
printf(95)

68 > = 95

· 95

Array

variable (4) →

2201	3201	4202	1205
80	40	50	60
a	b	c	d

array

1000	1001	1002	1003
80	40	50	60

Type array name [array size] =

```
(int) int arr[5]={4,6,7,2,3};
```

$$n=5$$

$$n-1$$

↑ ↑ ↑ ↑  
0 1 2 3 4

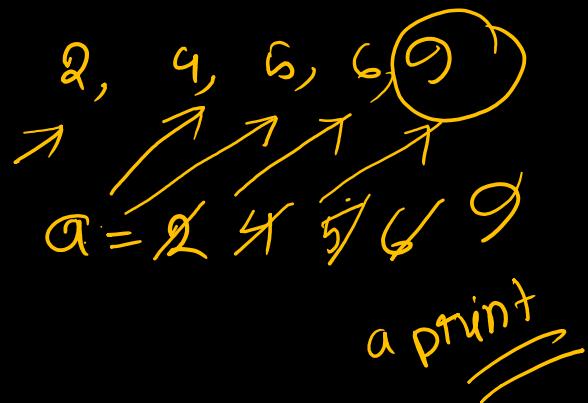
int a=arr[0]



$$4$$

0 →  
size-1

maximum value



Find the maximum value from array

```
printf("Enter the element of array:\n");
{ for(int i=0 ; i<5;i++){
    scanf("%d ",&arr[i]);
}
```

```
int a=arr[0];
for(int i=0 ; i<5 ; i++){
    if(a<arr[i]){
        a=arr[i];
    }
}
```

---

```
printf("Maximum value is: %d",a);
```

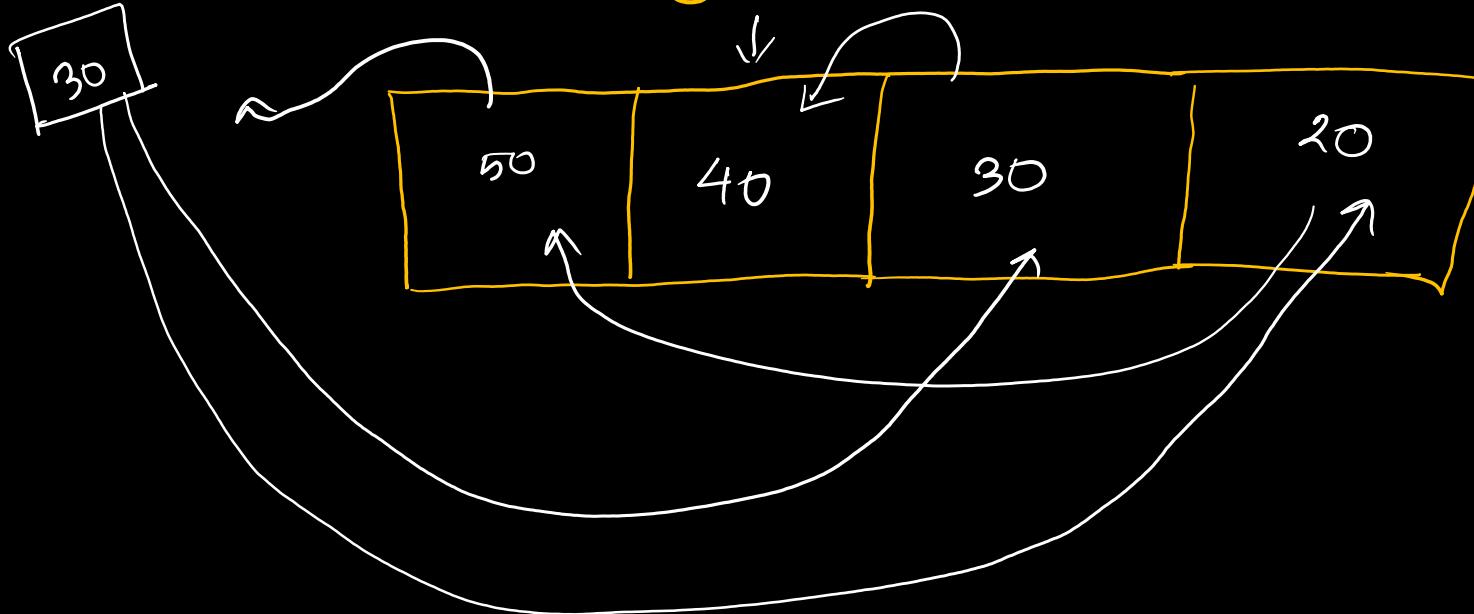
---

• Enter the element of array  
i = 0 → 1  
i = 1 → 2  
i = 2 → 3  
i = 3 → 4  
i = 4 → 5  
a = 1

Output 5

① How to use an array in function

② How to reverse an array without using a  
extra array with function



```

int main(){
    int arr[4] = {20,30,40,50};

    for(int i=0 ; i<4 ; i++){
        printf("%d ",arr[i]);
    }
    printf("\n");
    reverse(arr,4);
    printf("Array after reverse:\n");
    for(int i=0 ; i<4 ; i++){
        printf("%d ",arr[i]);
    }
}

```

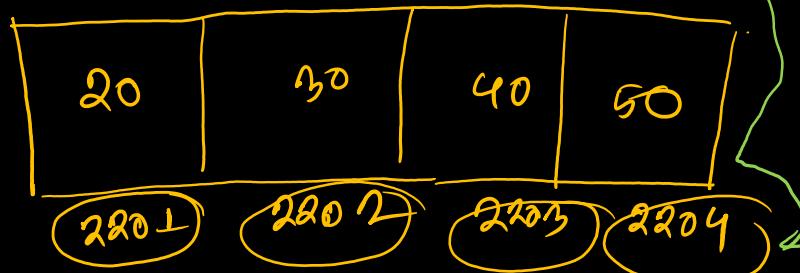
↙

```

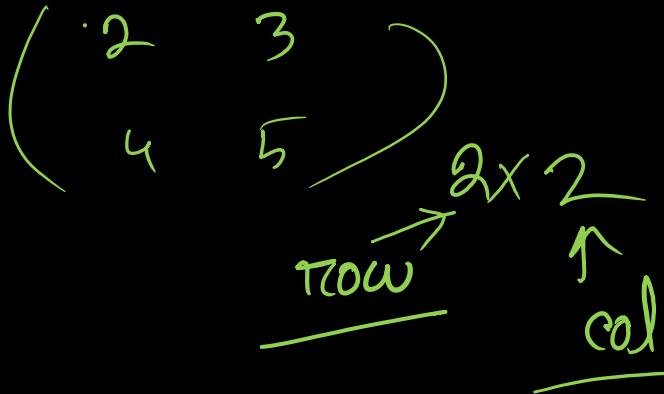
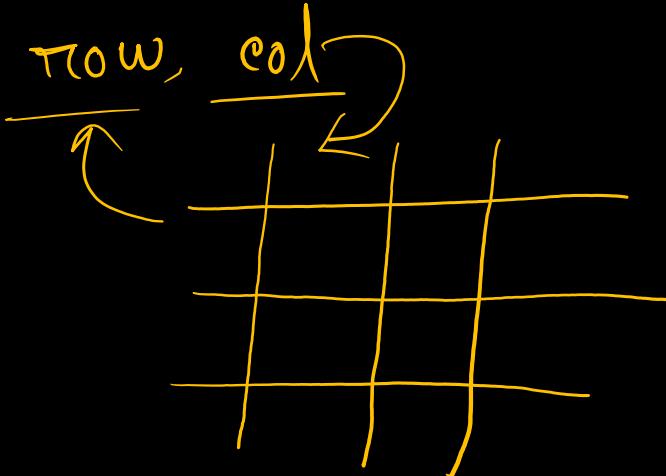
void reverse(int arr[],int a){
    int temp=0;
    int start=0;
    int end=a-1; ↗
    while(start<end){
        temp=arr[start];
        arr[start]=arr[end];
        arr[end]=temp;
        start++;
        end--;
    }
}

```

return 0;



## 2d array



data type name  $[] [ ] \nearrow 3$   
char arr [2] [3]  
 $0 \rightarrow 1 \downarrow 2 \downarrow 3 \}$   
 $1 \rightarrow 3 \quad 4 \quad 5 \}$

```

int main(){
    char arr[2][3]={{'e','m','o'},{'n','h','k'}};
    for(int i=0; i<2; i++){
        for(int j=0 ; j<3 ; j++){
            printf("%c ",arr[i][j]);
        }
        printf("\n");
    }
}

```

Diagram illustrating the execution flow:

- The outer loop (indicated by a vertical arrow) iterates over  $i$  (0 to 1).
- For each iteration of the outer loop, the inner loop (indicated by a horizontal arrow) iterates over  $j$  (0 to 2).
- Each iteration of the inner loop prints one character from the array.
- After each row (iteration of the outer loop), a new line is printed.
- Annotations show "Row" pointing to the outer loop iterations, "col" pointing to the inner loop iterations, and "2<sup>n</sup> Row" indicating the total number of rows (2).

 $2 \times 3$ 

1

outer loop  $\phi \perp 2$

inner loop  $\phi \perp 3$

arr [0] [1] [2]

(1) (2)

output

e m o  
n h k

```

#include<stdio.h>

int main(){
    int a[3][3];
    int b[3][3];
    int c[50][20];

    printf("Enter the first array:\n");

    for(int i=0 ; i<3 ; i++){
        for(int j=0 ; j<3 ; j++){
            scanf("%d",&a[i][j]);
        }
    }

    for(int i=0 ; i<3 ; i++){
        for(int j=0 ; j<3 ; j++){
            scanf("%d",&b[i][j]);
        }
    }
}

```

For loop analysis:

```

for(int i=0 ; i<3 ; i++){
    for(int j=0 ; j<3 ; j++){
        int sum=0;
        for(int k=0 ; k<3 ; k++){
            sum += a[i][k] * b[k][j];
        }
        c[i][j]=sum;
    }
}

```

$$C = A \times B$$

15	3+	

$$\begin{matrix} (0,0) & (0,1) & (0,2) \\ 3 & 2 & 1 \\ (1,0) & 3 & 1 \\ 4 & 5 & 6 \\ (2,0) & (2,1) & (2,2) \end{matrix} \quad 3 \times 3$$

(a)

$$\begin{matrix} (0,0) & (0,1) & (0,2) \\ 1 & 4 & 7 \\ (1,0) & 3 & 5 \\ 6 & 2 & 8 \\ (2,0) & (2,1) & (2,2) \end{matrix}$$

(b)

$$\frac{i=0 \leq 3}{j=0 \leq 3}$$

$$\frac{}{sum=0}$$

$$\frac{k=0 \leq 3}{i \quad \quad \quad k \quad \quad \quad j}$$

$$Sum = \underset{\emptyset}{Sum} + a[0][0] \times b[0][1]$$

$$c[0][0] = 0 +$$

```
for(int i=0 ; i<3 ; i++){
    for(int j=0 ; j<3 ; j++){
        printf("%d ",c[i][j]);
    }
    printf("\n");
}

return 0;
}
```

Char array ( string )

< string.h >

char arra [6] = { 'G', 'e', 'k', 'f', 'j', '\0' }  
                  ↙

char str [100] = "Geekfj" ↙  
                  ↙  
strlen( str )

str<sup>v</sup> = Em

strcpy ( str, 'Em' );

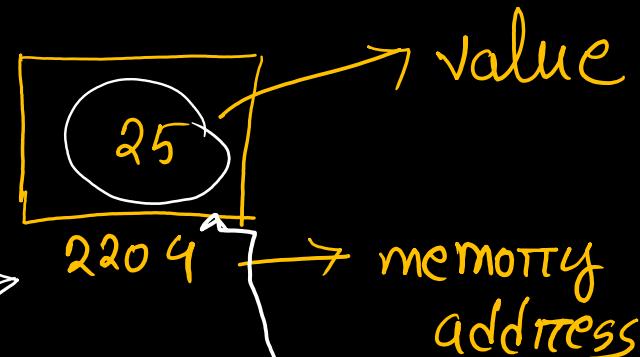
int a = strcmp ( str1, str2 );

T<sup>0</sup> ↘ ↗ 1  
f

Pointer → stores memory location

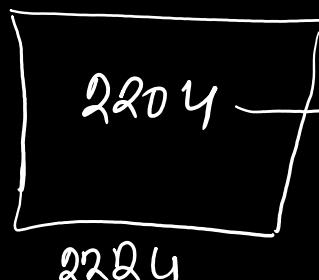
int a = 25;

int \*P = &a



value at  
address

address off



$$a = 25$$

$$*P = \&a$$

$$(" \% P ", \&a) \rightarrow 2204$$

$$(" \% P ", P) \rightarrow 2204$$

$$(" \% P ", \&P) \rightarrow 2200$$

$$(" \% d ", *P) \rightarrow \underline{\underline{26}}$$

25

2204

(a)

2204  
=

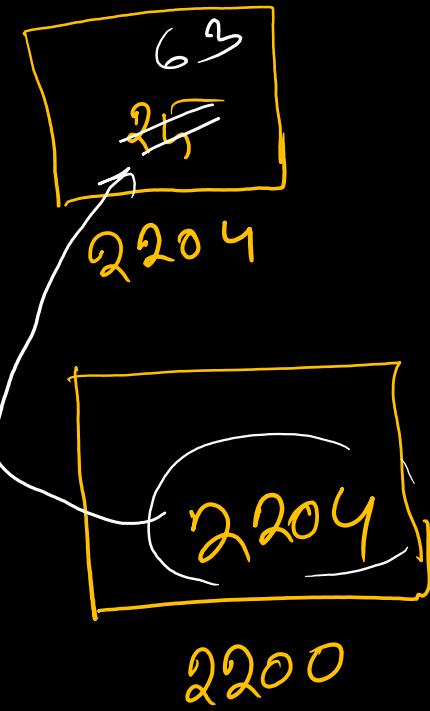
→ 2200  
P

$a = 25;$

$*p = \& a$

value at  
address

$*p = 63$



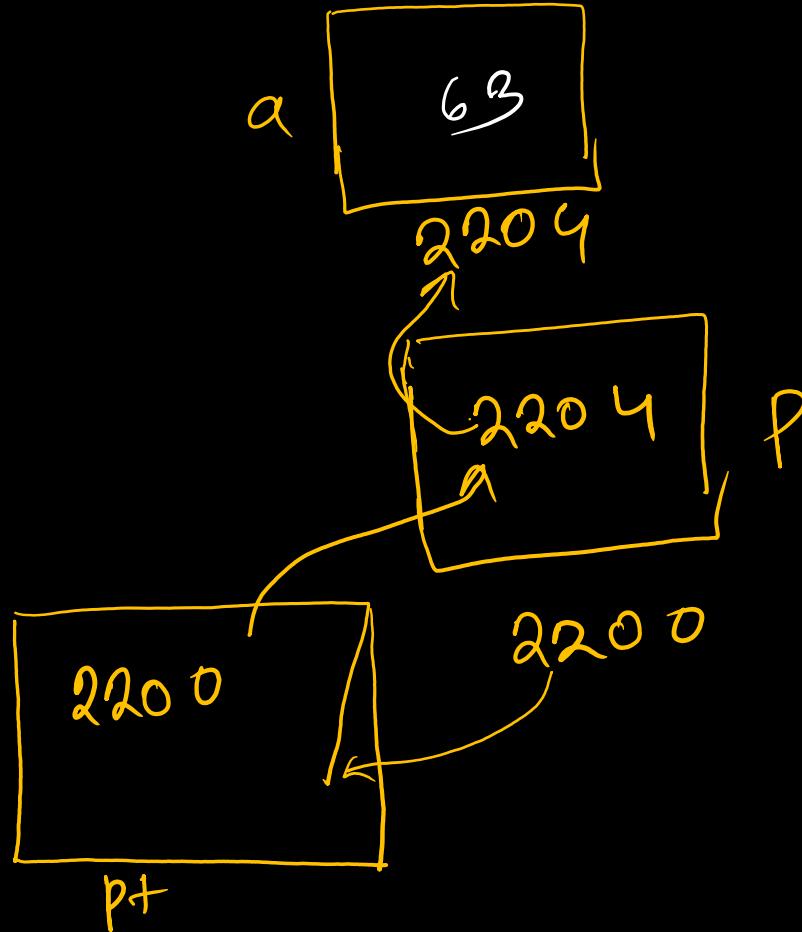
$$a = 25$$

$$\star P = \& a$$

$$\star\star p_t = \& P$$

$$\star\star p_t = 63;$$

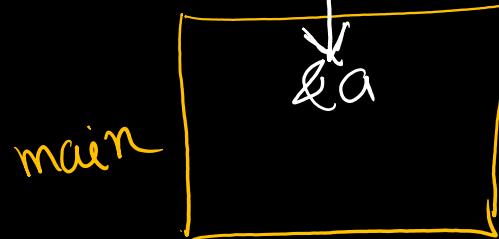
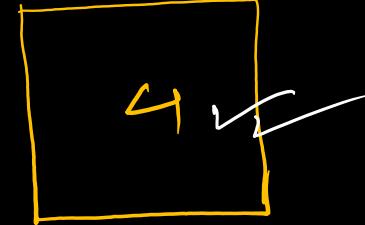
$$p_t = 63$$



```
void square(int *p){  
    *p=*p * *p;  
}
```

```
int main(){  
    int a=4;  
  
    printf("Before square:%d\n",a);  
    square(&a);  
    printf("After square:%d\n",a);
```

16



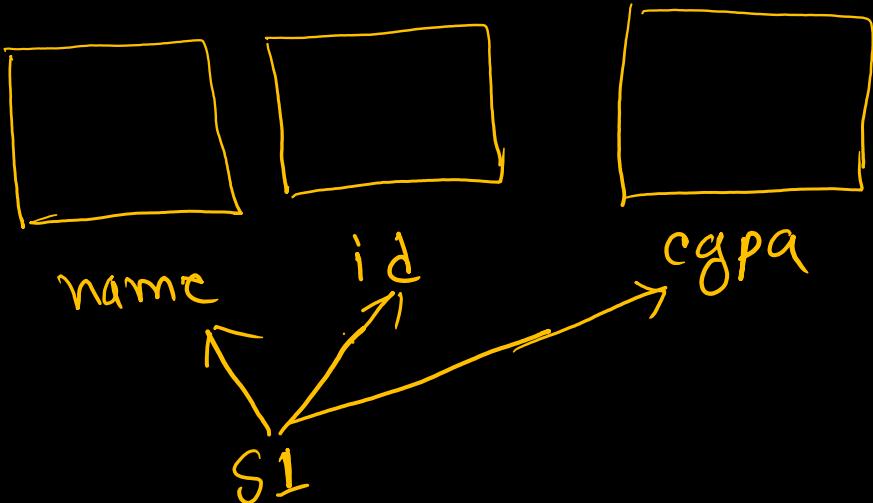
## Structure

int, float, char      int = ~~3.8~~, 'A'

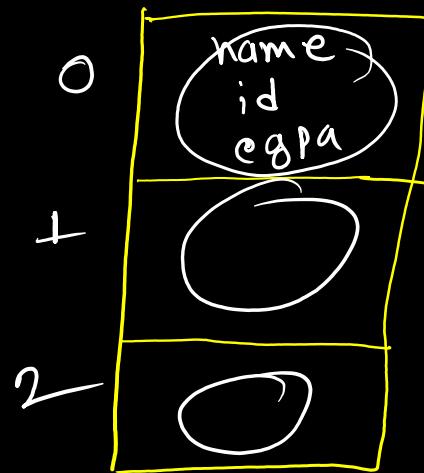
Struct name {  
    char name [100];  
    int id;  
    float cgpa;  
};

✓

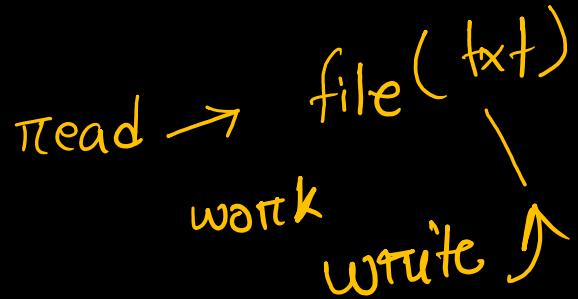
```
struct Student{  
    char name[100];  
    int id;  
    float cgpa;  
};
```



$S2 [3]$



File



FILE \*ptr;

$\downarrow$   
open  $\xleftarrow{w}$   $\xleftarrow{R}$  } close

## Part-II Question Practice

1. manual tracing
  2. Output
  3. problem
- calculator
- 

## Manual tracing (Q1)

Show manual tracing of the following code segment and find output

```
int num, sum, rem;  
num=3459; ✓  
sum=0; ✓  
while(num>0){  
    rem=num%10;  
    sum=sum+rem;  
    num=num/10;  
    if (num==0) &&(sum%10 !=sum){  
        printf("%d %d %d\n", rem, sum, num);  
        num=sum;  
    }  
}
```

- rem = 3 ✓
- sum = 21 ✓
- num = 0 ✓

10 3459 | 345  
3450 | 9 → rem

$$num = 3459$$

$$sum = 0 + 9$$

$$rem = 9$$

1st iteration (3459 > 0)

$$\begin{aligned} rem &= 3459 \% 10 \\ &= 9 \end{aligned}$$

$$\begin{aligned} sum &= 0 + 9 \\ &= 9 \end{aligned}$$

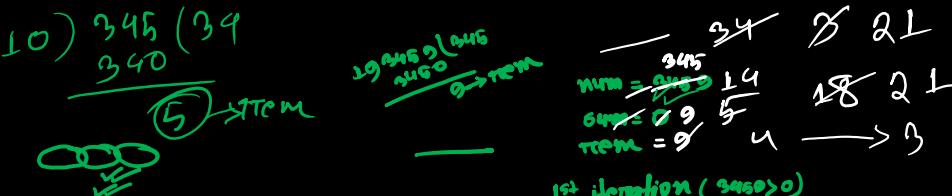
$$num = 3459 / 10$$

$$\begin{aligned} &= 345 \\ \text{condition } (345 &= 0) &\text{ F/T} \end{aligned}$$

```

Show manual tracing of the following code segment and find output
int num, sum, rem;
num=3459;
sum=0;
while(num>0){
    rem=num%10;
    sum=sum+rem;
    num=num/10;
    if(num==0) &&(sum%10 !=sum){
        printf("%d %d %d\n", rem, sum, num);
        num=sum;
    }
}

```



2nd iteration ( $345 > 0$ ) - T

$$\begin{aligned}
 \text{rem} &= 345 \% 10 \\
 &= 5 \\
 \text{sum} &= 0 + 5 = 14 \\
 \text{num} &= 345 / 10 = 34 \\
 \text{condition } (34 &\neq 0) &\text{ T}
 \end{aligned}$$

3rd iteration

$$\begin{aligned}
 \text{rem} &= 34 \% 10 \\
 &= 4 \\
 \text{sum} &= 14 + 4 = 18 \\
 \text{num} &= 34 / 10 \\
 &= 3 \\
 \text{condition } (3 &\neq 0) &\text{ T}
 \end{aligned}$$

4th iteration

$$\begin{aligned}
 \text{rem} &= 3 \% 10 \\
 &= 3 \\
 \text{sum} &= 18 + 3 = 21
 \end{aligned}$$

$$\begin{aligned}
 \text{num} &= 3 / 10 \\
 &= 0
 \end{aligned}$$

Condition:  $0 = 0$  and  $(21 \% 10) = 21$

printf (rem, sum, num)

num = 21

output

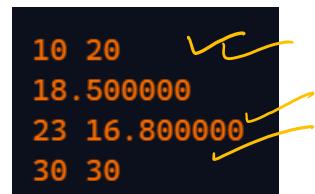
- item = 3
- sum = 21
- num = 0

✓

## Manual tracing (Q2)

Show manual tracing for the following program and find output

```
#include<stdio.h>
int a, b;
int func1(float x);
void func2(int x, float y);
int main(){
    a=10;
    b=20;
    printf("%d %d\n", a, b);
    a=func1(18.5);
    func2(23, 16.8);
    printf("%d %d\n", a, b);
    return 0;
}
int func1(float x) {
    b=b+a;
    printf("%f\n", x);
    return b;
}
void func2(int x, float y){
    printf("%d %f\n", x, y);
    return;
}
```



```
10 20 ✓
18.500000
23 16.800000
30 30
```

Show manual tracing for the following program and find output

```
#include<stdio.h>
int a, b;
int func1(float x);
void func2(int x, float y);
int main()
{
    a=10;
    b=20;
    printf("%d %d\n", a, b);
    a=func1(18.5);
    func2(23, 16.8);
    printf("%d %d\n", a, b);
    return 0;
}
int func1(float x)
{
    b=b+a;
    printf("%f\n", x);
    return b;
}
void func2(int x, float y)
{
    printf("%d %f\n", x, y);
    return;
}
```

$$a = 10$$

$$b = 20$$

printf(a, b)

a = function call(18.5)

10 20  
18.5  
23 16.8  
30, 30

$$a = 10 \quad 30$$
$$b = 30$$

x  
y  
func2(23, 16.8)

printf(x, y)  
return

func1(18.5)

$$\begin{aligned} b &= b+a \\ &= 20 + 10 \\ &= 30 \end{aligned}$$

printf(x)

return b

function call  
func2(23, 16.8)

printf(a, b);

output

10 20  
18.5  
23 16.8  
30 30

## Manual tracing (Q3)

Show manual tracing of the following code segment and find output.

```
char str1[7]={'\0'};  
char str2[4]={'\0'};  
int i, k;  
strcpy(str1, "CSE");  
strcpy(str2, "UIU");  
i=strlen(str1);  
for(k=0; str2[k] !='\0'; ++k)  
    str1[i+k]=str2[k];  
puts(str1);  
printf("\n");  
strrev(str1);  
puts(str1);
```

CSEUIU  
UIUESC ↴

Show manual tracing of the following code segment and find output.

```
char str1[7]={'\0'};  
char str2[4]={'\0'};  
int i, k;  
strcpy(str1, "CSE");  
strcpy(str2, "UIU");  
i=strlen(str1);  
for(k=0; str2[k]!='\0'; ++k)  
    str1[i+k]=str2[k];  
puts(str1);  
printf("\n");  
strrev(str1);  
puts(str1);
```

$$\begin{aligned} \text{str1} &= \text{null} \\ \text{str2} &= \text{null} \end{aligned}$$

i, k

$$\begin{aligned} \text{str1} &= \underline{\text{CSE}} \\ \text{str2} &= \underline{\text{UIU}} \\ i &= 3 \end{aligned}$$

$$\text{str1} = \underline{\text{CSE}} \cup \underline{\text{I}} \cup \underline{\text{U}}$$

$$\begin{array}{|c|c|} \hline \text{print(str1)} & \text{print(str1)} \\ \text{reverse(str1)} & \\ \text{str1} = \text{UIUESC} & \\ \hline \end{array}$$

.CSEVIU  
.VIUESC

output

1st iteration ( $\text{str1}[0] \neq '\text{I}'$ ) T  
 $k=0$

$$\begin{array}{|c|c|} \hline \text{str1}(3+0) = \text{str2}[0] & \\ \text{str1} = \underline{\text{CSEU}} & \\ \hline \end{array}$$

3rd iteration  
( $\text{str1}[2] \neq \text{null}$ ) T

$$\begin{array}{|c|c|} \hline \text{str1}(3+2) = \text{str2}[2] & \\ = \text{U} & \\ \text{str1} = \text{CSEU}\text{U} & \\ \hline \end{array}$$

2nd iteration ( $\text{str2}[1] \neq \text{null}$ ) T  
 $k=1$

$$\begin{array}{|c|c|} \hline \text{str1}(3+1) = \text{str2}[1] & \\ = \text{I} & \\ \text{str1} = \text{CSEUI} & \\ \hline \end{array}$$

$k=2$

4th iteration  
 $\text{str2}[3] = \text{null}$   
End

## Manual tracing (Q4)

```
#include<stdio.h>
void change (int *x, int *y, int z) {
    *x=*x+10;
    *y=*y+3;
    z=z+5;
    return;
}
int main(){
    int a=LAST_THREE_DIGIT_OF_YOUR_STUDENT_ID%5;
    int b=LAST_THREE_DIGIT_OF_YOUR_STUDENT_ID/5;

    int c= LAST_THREE_DIGIT_OF_YOUR_STUDENT_ID;
    printf("%d %d %d\n", a, b, c);
    change(&a, &b, c);
    printf("%d %d %d\n", a, b, c);
    return 0;
}
```

```

#include<stdio.h>
void change (int *x, int *y, int z) {
    *x = *x + 10;
    *y = *y + 3;
    z = z + 5;
    return;
}
int main(){
    int a=LAST_THREE_DIGIT_OF_YOUR_STUDENT_ID%5;
    int b=LAST_THREE_DIGIT_OF_YOUR_STUDENT_ID/5;
    -
    int c= LAST_THREE_DIGIT_OF_YOUR_STUDENT_ID;
    printf("%d %d %d\n", a, b, c);
    change(&a, &b, c);
    printf("%d %d %d\n", a, b, c);
    return 0;
}

```

$$a = 123 \% 5 \rightarrow 3$$

$$b = 125 / 5 \rightarrow 24$$

$$c = 125$$

printf (a, b, c)

function call

↓  
change (&a, &b, c)

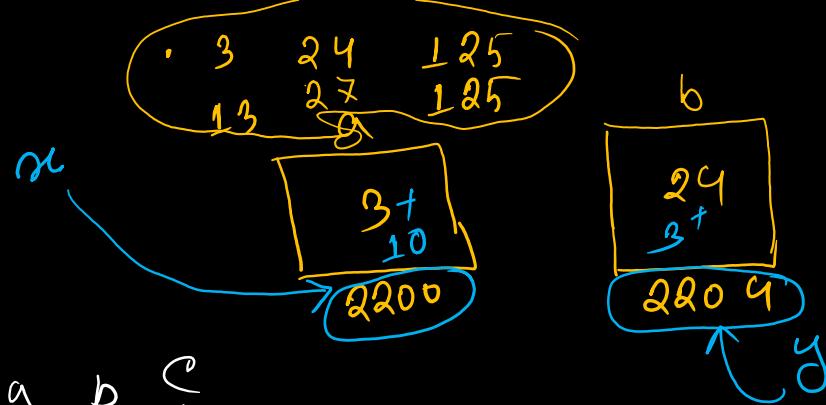
123  
5)  $\frac{123}{10} (24)$   
 $\frac{23}{20}$   
 $\frac{3}{0} \Rightarrow$

a ↓ b ↓ c ↓  
fun. change (\*x, \*y, z)

$$\begin{aligned} *x &= *x + 10 \\ &= 3 + 10 \rightarrow 13 \end{aligned}$$

$$\begin{aligned} *y &= *y + 3 \\ &= 24 + 3 \rightarrow 27 \end{aligned}$$

$$\begin{aligned} z &= 125 + 5 \rightarrow 120 \\ \text{return}; \end{aligned}$$



printf (a, b, c)  
output

• 3 24 125  
• 13 27 125

## Manual tracing (Q5)

```
#include <stdio.h>
#include <string.h>
int main(){
    char str1[50] = "CSE-1111 SPL";
    char str2[50] = "I am a UIUian";

    int i = strlen(str1) * 0.5 - 2;

    for (int m = 0; i + m < strlen(str1); m += 3) {
        str1[i + m] = str2[m];
    }

    strcat(str1, str2);

    if (strcmp(str2, str1) > 0) {
        strncat(str1, "CSE is awesome.", 6);
    }
    else{
        strncat(str2, "CSE is awesome.", 6);
    }
    return 0;
}
```

C Code for 2(a)

```

#include <stdio.h>
#include <string.h>
int main(){
    char str1[50] = "CSE-1111 SPL";
    char str2[50] = "I am a UIUian";

    int i = strlen(str1) * 0.5 - 2;
    for (int m=0; i+m < strlen(str1+1); m+=3) {
        str1[i+m] = str2[m];
    }
    strcat(str1, str2);

    if (strcmp(str2, str1) > 0) {
        strncat(str1, "CSE is awesome.", 6);
    } else {
        strncat(str2, "CSE is awesome.", 6);
    }
    return 0;
}

```

C Code for 2(a)

$i = 0$

$m = 0$

1st iteration

$$m=0 \quad i+m < \text{strlen}(str1) \\ (0+0 < 12) \quad T$$

$$str1[4+0] = str2[0]$$

$$str1 = CSE-I\cancel{L}L1 SPL$$

$m = 3$

2nd iteration ( $m=3$ )

$$i+m < \text{strlen}(str2) \quad (T) \\ 7 < 12$$

$$str1[7] = str2[3]$$

$$str1 = CSE-I11m SPL$$

$$m = 3 + 3 \rightarrow 9$$

$$str1 = \begin{matrix} C & S & E & - & I & L & L & 1 & S & P & L \\ 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 \end{matrix}$$

$$str2 = \begin{matrix} I & a & m & a & U & I & U & i & a & n \\ 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 \end{matrix}$$

$$\begin{aligned} i &= \text{strlen}(str1) * 0.5 - 2 \\ &= (12 * 0.5) - 2 \\ &= 6 - 2 \end{aligned}$$

```

#include <stdio.h>
#include <string.h>
int main(){
    char str1[50] = "CSE-1111 SPL";
    char str2[50] = "I am a UIUian";

    int i = strlen(str1) * 0.5 - 2;
    for (int m=0; i+m < strlen(str1); m+=3){
        str1[i+m] = str2[m];
    }
    strcat(str1, str2);
    if(strcmp(str2, str1) > 0){
        strncat(str1, "CSE is awesome.", 6);
    } else{
        strncat(str2, "CSE is awesome.", 6);
    }
    return 0;
}

```

C Code for 2(a)

3rd iteration (m=9)

$$\frac{4+9}{13} < 12 \text{ T}$$

loop end

strcat(str1, str2)

str1 = CSE-1111 SPL  
I am a UIUian

CSE  $\rightarrow$  awesome

↑ 1 1 1 1 1  
↓ 2 3 4 5 ↘

$i=6$        $m=0$

{

1st iteration  
 $m=0$        $i+m < \text{strlen(str1)}$   
 $(9+0 < 12)$  T  
 $\text{str1}[4+0] = \text{str2}[0]$   
 $\text{str1} = \text{CSE-1111 SPL}$   
 $m=3$

Str1 = CSE-1111 SPL  
~~0 1 2 3 4 5 6 7 8 9 10 11 12~~  
~~I am a UIUian~~  
 $i = \text{strlen(str1)} * 0.5 - 2$   
 $= (12 * 0.5) - 2$   
 $= 6 - 2$   
 $= 4$

2nd iteration (m=9)  
 $i+m < \text{strlen(str1)}$  (T)  
 $(4+9 < 12)$   
 $\text{str1}[4] = \text{str2}[9]$   
 $\text{str1} = \text{CSE-1111 SPL}$   
 $m=9+3 \rightarrow 12$

condition (  $\frac{\text{strcmp(str2, str1)} > 0}{\text{return 1}}$  ) T

str1 = CSE-1111 SPL am a UIUian CSE is

## Manual tracing (Q6)

```
#include<stdio.h>
int main(){
    FILE *fp1;
    int i, sum;
    int num[5]={0};
    num[3]= LAST_THREE_DIGIT_OF_YOUR_STUDENT_ID%10;
    num[0]= LAST_THREE_DIGIT_OF_YOUR_STUDENT_ID%100;
    num[4]= LAST_THREE_DIGIT_OF_YOUR_STUDENT_ID%1000;
    num[1]=num[0]+num[3];
    num[2]=num[1]+num[4];
    fp1= fopen("D:\\students\\dest.txt", "w");
    sum=0;
    for(i=4; i>=0; i--){
        if(A[i]%2==0){
            sum=sum+num[i];
            fprintf(fp1, "%d\\n", num[i]);
        }
    }
    fprintf(fp1, "%d", sum);
    fclose(fp1);
    return 0;
}
```

```
#include<stdio.h>
int main(){
    FILE *fp1;
    int i, sum;
    int num[5]={0};
    num[3]=LAST_THREE_DIGIT_OF_YOUR_STUDENT_ID%10;
    num[0]=LAST_THREE_DIGIT_OF_YOUR_STUDENT_ID%100;
    num[4]=LAST_THREE_DIGIT_OF_YOUR_STUDENT_ID%1000;
    num[1]=num[0]+num[3];
    num[2]=num[1]+num[4];
    fp1=fopen("D:\\students\\dest.txt", "w");
    sum=0;
    for(i=4; i>=0; i--){
        if(A[i] % 2 == 0){
            sum=sum+num[i];
            fprintf(fp1, "%d\\n", num[i]);
        }
    }
    fprintf(fp1, "%d", sum);
    fclose(fp1);
    return 0;
}
```

\* fp1  
 i sum  
 $num[5] = \{0\}$   
 $num[3] = 123 \% 10$   
 $= 3$   
 $num[0] = 123 \% 100$   
 $= 23$   
 $num[4] = 123 \% 1000$   
 $= 123$

123  
 10) 123 (12  
 120  
 3

$$num[1] = num[0] + num[3]$$

$$= 23 + 3 \\ = 26$$

$$num[2] = num[1] + num[4]$$

$$= 26 + 123 \\ = 149$$

open file

$$sum = 0$$

1st iteration

$$(4) = 0 \quad 123 \% 2 \neq 0$$

condition ( $num[4] \% 2 = 0$ )  
 F

$$i = 4 - 1 = 3 \\ \text{loop end}$$

2nd iteration

$$i \geq 0 (3 \geq 0) \quad T$$

condition ( $num[3] \% 2 = 0$ )  
 F

$$i = 3 - 1 = 2$$

```
#include<stdio.h>
int main(){
    FILE *fp1;
    int i, sum;
    int num[5]={0};
    num[3]=LAST_THREE_DIGIT_OF_YOUR_STUDENT_ID%10;
    num[0]=LAST_THREE_DIGIT_OF_YOUR_STUDENT_ID%100;
    num[4]=LAST_THREE_DIGIT_OF_YOUR_STUDENT_ID%1000;
    num[1]=num[0]+num[3];
    num[2]=num[1]+num[4];
    fp1=fopen("D:\\students\\dest.txt", "w");
    sum=0;
    for(i=4; i>=0; i--){
        if(A[i]>=0 & i<=0){
            sum=sum+num[i];
            fprintf(fp1, "%d\n", num[i]);
        }
    }
    fprintf(fp1, "%d", sum);
    fclose(fp1);
    return 0;
}
```

26  
3rd iteration ( $i \geq 0$ ) (T)

condition ( $num[i] \neq 0$ )  
(T)  
 $i = 2 - 1 = 1$

4th iteration ( $i \geq 0$ ) (T)

condition ( $num[i] \neq 0$ )  
(T)

$$sum = 0 + num[1]$$

$= 26$   
write in file ( $num[i]$ )  
 $i = 1 - 1 = 0$

26  
sum = 26

\* fp1  
i sum  
 $num[5] = 90$   
 $num[3] = 123 \% 10$   
 $= 3$   
 $num[0] = 123 \% 100$   
 $= 23$   
 $num[4] = 123 \% 1000$   
 $= 123$

$$\begin{aligned} num[1] &= num[3] + num[5] \\ &= 23 + 3 \\ &= 26 \\ num[2] &= num[3] + num[4] \\ &= 26 + 123 \\ &= 149 \end{aligned}$$

124  
10 123 12  
100  
2  
3

open file

sum=0

get iteration  
( $i \geq 0$ )  
condition ( $num[i] \neq 0$ )  
(T)  
 $i = 9 - 1 = 8$   
loop end

and iteration  
( $i \geq 0$ )  
condition ( $num[i] \neq 0$ )  
(T)  
 $i = 9 - 1 = 8$

5th iteration ( $i \geq 0$ ) (T)

condition ( $num[i] \neq 0$ )  
(T)  
 $i = -1$

6th iteration ( $-1 \geq 0$ ) (R)  
loop end

write in file (sum)  
26

file (close)

file

- 26
- 26

## Output (Q1)

```
#include<stdio.h>
int main(){
    int b = LAST_FOUR_DIGIT_OF_YOUR_STUDENT_ID % 11;
    int a[5] = {b+1, b+2, b+3, b+4, b+5};
    int *p1,t,u,v,w;

    p1=a;
    t= (*p1)++;
    u = *p1;
    v = *++p1;
    w = *(++p1);

    printf("%d %d %d %d", t, u, v, w);
}
```

```

#include<stdio.h>
int main(){
    int b = LAST_FOUR_DIGIT_OF_YOUR_STUDENT_ID % 11;
    int a[5] = {b+1, b+2, b+3, b+4, b+5};
    int *p1, t, u, v, w;
    p1=a;
    t=(*p1)++;
    u=*p1;
    v= *++p1;
    w=*(++p1);
    printf("%d %d %d %d", t, u, v, w);
}

```

$++a \rightarrow$  pre-increment  
 $a++ \rightarrow$  post-increment

$a = 5$

$b = ++a$        $b = 5 \rightarrow a = 6$

$b = 6$

$$b = 2$$

$$a[5] = \{ b+1, b+2, b+3, b+4, b+5 \}$$

$$= \{ 3, 4, 5, 6, 7 \}$$

$*p1, t, u, v, w$

$$p1 = a[0]$$

$$t = (*p1)++$$

$$t = 3$$

$$u = *p1(a[0])$$

$$= 4 \quad (a[0](1))$$

$$v = *++p1 \quad v = 4$$

$$w = *(++p1) \quad w = 5$$

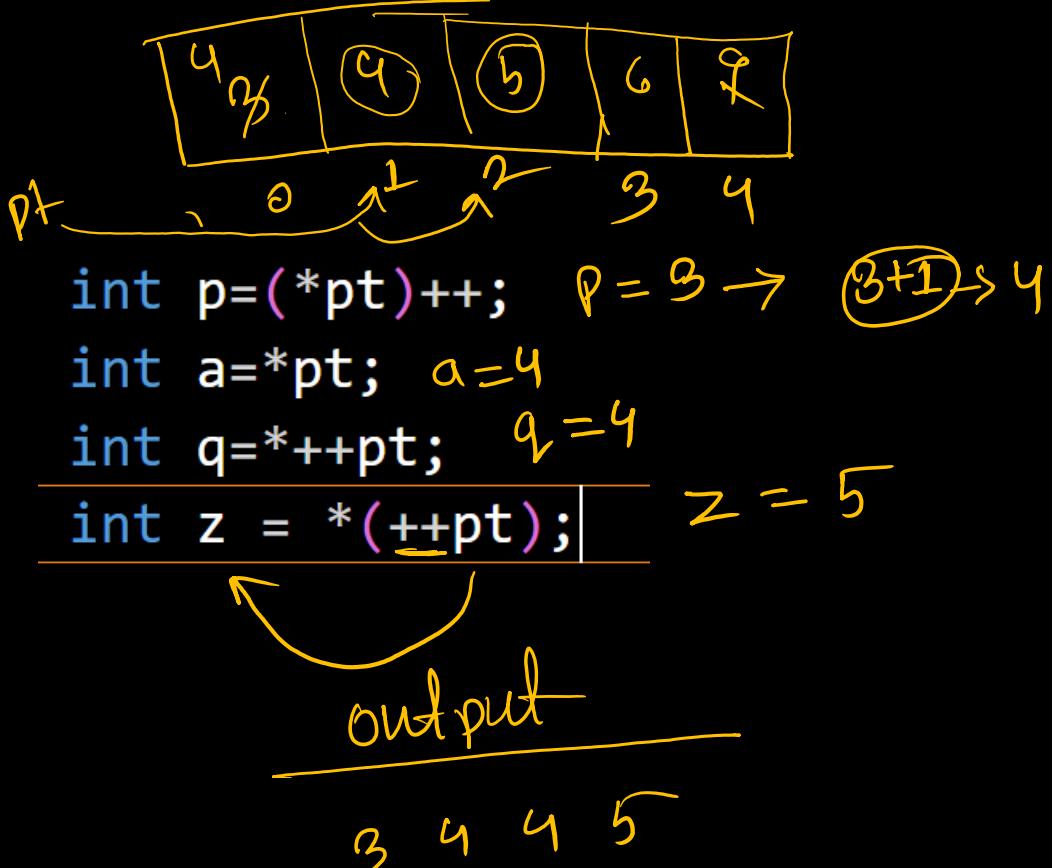
```

#include<stdio.h>
int main(){
    int b = LAST_FOUR_DIGIT_OF_YOUR_STUDENT_ID % 11;
    int a[5] = {b+1, b+2, b+3, b+4, b+5};
    int *p1,t,u,v,w;

    p1=a;
    t=(*p1)++;
    u=*p1;
    v= *++p1;
    w=*(++p1);

    printf("%d %d %d %d", t, u, v, w);
}

```



## Output (Q2)

Write the output of the following code segment. The ASCII code of 'A' is 65 and 'a' is 97.

```
#include<stdio.h>
void function(char str[], int code, int n);
void main() {
    char name[50] = "Abu Sayeed Shiblu";
    function(name, 'A', 0);
}
void function(char str[], int code, int n) {
    printf("%c, ", code);
    int num = code % strlen(str);
    if (n > 4) return;
    function(str, str[num], ++n);
}
```

A

— — — —

Write the output of the following code segment. The ASCII code of 'A' is 65 and 'a' is 97.

```
#include<stdio.h>
void function(char str[], int code, int n);
void main() {
    char name[50] = "Abu Sayeed Shibli";
    function(name, 'A', 0);
}
void function(char str[], int code, int n) {
    printf("%c, ", code);
    int num = code % strlen(str);
    if (n > 4) return;
    function(str, str[num], ++n);
}
```

length = 17

name[50] = Abu Sayeed Shibli

0 12 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17

function call (name, 'A', 0);

Write the output of the following code segment. The ASCII code of 'A' is 65 and 'a' is 97.

```
#include<stdio.h>
void function(char str[], int code, int n);
void main() {
    char name[50] = "Abu Sayeed Shibli";
    function(name, 'A', 0);
}
void function(char str[], int code, int n) {
    printf("%c", code);
    int num = code % strlen(str);
    if (n > 4) return;
    function(str, str[num], ++n);
}
```

• A  
• 10

## Iteration 1

(char str[], code, n)

printf(code)

num = 65 % 17 → 14

condition (14 > 4) return (F)

function call (str, str[num], ++n)  
(b) 1

length = 17      ↗ 8  
name = Abu Sayeed Shibli      ↗ 11 ↗ 14  
str(13) = 'i'      ↗ 9 ↗ 10 ↗ 11 ↗ 12 ↗ 13 ↗ 14  
↓ 105

function (name, 'A', 0);

iteration and

(str, 98, 1)

printf(code)

num = 98 % 17

= 13

condition (13 > 4) (F)

function call (str, 'i', 2)

A b



Iteration 1

char str[], code, n  
print(code)  
num = 65 % 12 → 14  
condition (n > 4) return(F)  
function call (str, str[num], t+n)

function (name, 'A', 0);

Iteration and  
(str, 98, 1)  
print(code)  
num = 98 % 12  
= 10  
condition (1 > 4) (F)  
function call (str, 1, 2)

4th iteration

(str, 32, 3)

print(code)

num = 32 % 12  
= 10

condition (3 > 4) (F)

function (str, 'l', 4)

str(3) = —  
as (A b i - l)

Output

(A b i - l)

3rd iteration

(str, 105, 2)

print(code)

num = 105 % 12  
= 3

condition (2 > 4) (F)

function call (str, -, 3)

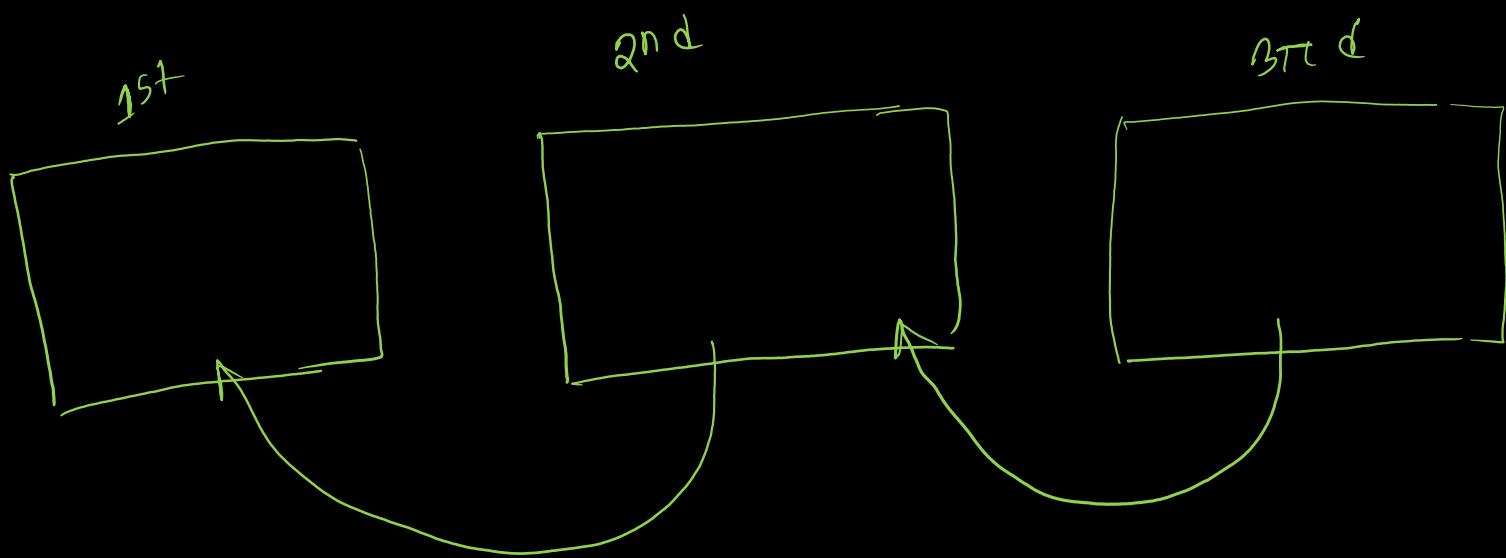
5th iteration

(str, 108, 9)

print(code)

num = 108 % 12  
= 6

condition (4 > 4) (T)  
return



1 - i  $\nabla A$

## Output (Q3)

```
#include <stdio.h>
int ara[5], x = 20;
void change(int p) {
    --p;
    p--;
}
void update(int n) {
    for(int i = n - 1; i >= 0; i--){
        ara[i] -= x;
        change(x);
    }
}
void main() {
    int n = 5;
    for(int i = 0; i < n; i++){
        ara[i] = (i + 5) * 2;
    }
    update(n);
    for(int i = 0; i < n; i++){
        printf("%d, ", ara[i]);
    }
}
```

C Code for 1(b)

## Code writing (Q1)

The dairy milk company **Farm-Fresh** needs your help to compute the **weight and average** daily milk production for each of their cows. The **amount of milk** for each individual cow, **m** is computed as follows:

$$m = aw^b, \text{ where, } w \text{ is calculated as follows:}$$

$$w = c(1 - e^{-dx}), \text{ where } x \text{ is the age of the cow in years, and } a, b, c \text{ and } d \text{ are coefficients.}$$

In your program,

- i) Write a function that takes the **age (x)**, and the **coefficients (a, b, c, d)** as parameters and compute the **weight and milk production** of a cow. The function doesn't return anything. It can take any other parameters as needed.
- ii) In the **main** function, call the above function and display the weight and the average milk production of a cow using the following data: **x=35, a = 0.87, b = 0.45, c = 800, d = 3.5**. Use the concept of **call by reference**.

$$m = aw^b$$
$$w = c(1 - e^{-dm})$$

exp

i) function  $\rightarrow$  age( $n$ ),  $a, b, c, d$

ii) main  $\rightarrow$  function  $\rightarrow$  call

```
void calculate_wv(float*m, float*w, float*x, float *a ,float*b, float*c, float*d){
```

```
    float f = *d * *x;  
    *w = *c * (1-exp(-f));  
    *m=*a * pow(*w,*b);  
}
```

```
int main(){  
    float m=0;  
    float w=0;  
  
    float x=35;  
    float a=0.87,b=0.45,c=800,d=3.5;  
  
    calculate_wv(&m,&w,&x,&a,&b,&c,&d);  
    printf("%f\n",m);  
    printf("%f",w);  
  
    return 0;
```

```
}
```

## Code writing (Q2)

Suppose, Karim is working as an informer of the Bangladesh Police. He sometimes needs to send important messages to the head office. In order to send a secret message, he has to ensure the security of the message to not make it available to anyone else but the boss. Therefore, instead of sending the message directly, Karim is trying to **change the message** in such a way that no one will be able to read it even if they become successful to hack it. To help Karim, we need to **develop a system** that helps to **change the message**. In the initial stage of building such a system, we want to **replace only the consonants** of the message by the **next consonants** that come in the **alphabet sequence from A to Z**. For example, replace B with C, etc. For Z, replace it with A.

Now, write a C program that will change every consonant present in the message to the next one.

tent — consonant

B → C  
E → F  
Z → A

```
int main(){

    char mg[100];

    printf("Input the text:");
    scanf("%[^\\n]s" , &mg);

    for(int i=0 ; mg[i] != '\0' ; i++){  

        if(mg[i] == 'Z'){
            mg[i]='A';
        }

        if((mg[i]>='A' && mg[i]<='Z') && !(mg[i]=='A' || mg[i]=='E'||mg[i]=='I' ||mg[i]=='O' || mg[i]=='U' )){  

            mg[i]++; // mg[i] = mg[i]+1;  

        }

        if(mg[i] == 'z'){
            mg[i]='a';
        }

        if((mg[i]>='a' && mg[i]<='z') && !(mg[i]=='a' || mg[i]=='e'||mg[i]=='i' ||mg[i]=='o' || mg[i]=='u' )){  

            mg[i]++; // mg[i] = mg[i]+1;  

        }
    }

    printf("Changed msg:%s",mg);
}
```

## Code writing (Q3)

Write a program that (i) declares a string str\_a (of size (LAST\_FOUR\_DIGITS\_OF\_YOUR\_STUDENT\_ID % 11 + 30)) and initializes with "**Your own name, Your own student id**". (ii) Take input from user into str\_a. This string may have alphabets and digits. (iii) Store only the numerical characters of str\_a into another string str\_b. (iv) If the string str\_b doesn't contain any numerical character, print 0 (zero), otherwise, print str\_b. Some example input/outputs are given below:

	Example 1	Example 2	Example 3	Example 4
Input	123	abc	123abe	12ab34
Output	123	0 ✓	123	1234

str-a → name + id  
str-b = ↴  
if str-b  
str-b = id  
0  
doesnt contain a number  
print(0)

```
#include<stdio.h>
#include<string.h>

int main(){
    char str_a[6984%11+30];
    int index=0;
    char str_c[6984%11+30];

    printf("Enter your name with id:");
    scanf("%[^\\n]s",&str_a);

    for(int i=0 ; str_a[i] != '\\0' ; i++){
        if(str_a[i]>='0' && str_a[i]<='9'){
            str_c[index++]=str_a[i];
        }
    }

    str_c[index]= '\\0';

    if(index == 0){
        printf("%d",0);
    }
    else{
        printf("%s",str_c);
    }
}
```