FUNDAMENTAL

PROGRAMING

21/11/2019 Fun Pm 12321

⇒ Program of Palindrom Number:

e.g (1)

{(1221) = 1x1000 + 2x100 + 2x10 + 1x1}

OR

{(1221) = 1x1000 + 2x100 + 2x10 + 1x1}

e.g (2)

{(1234) = 1x1000 + 2x100 + 3x10 + 4x1}

OR

{(4321) = 4x1000 + 3x100 + 2x10 + 1x1}

$$10 \overline{)1234}\ 123 \quad (1230,\ a = 4)$$

$$10 \overline{)123}\ 12 \quad (120,\ b = 3)$$

$$10 \overline{)12}\ 1 \quad (10,\ c = 2)$$

$$10 \overline{)1}\ 0 \quad (d = 1)$$

1234 = 1x1000 + 2x100 + 3x10 + 4x1

4321 = ax1000 + bx100 + cx10 + dx1

```
#include <stdio.h>
~~void~~
int main( )
{
    int num, a, b, c, d, reverse, res;
Printf ("Enter 4 digit number");
Scanf ("%d", & num);

a = num % 10 ;  ——> a stores reminder
res = num / 10 ;  ——> res stores quotient

b = res % 10;
res = res / 10;

c = res % 10;
res = res / 10;

d = res % 10;


reverse = a*1000 + b*100 + c*10 + d*1

If (reverse == num)
{ Printf ("Number is Palindrom"); }

else
{ Printf ("Number is not Palindrom"); }

}
```

=> Program

To find minimum number of notes

e.g 2019

notes {5000, 1000, 500, 200, 50, 20, 10, 5, 2, 1}

1000 + 1000 + 10 + 5 + 2 + 2

=> $\frac{2019}{1000} = 2.019$

2 (integer)

(2019 - 2x1000 = 019)

=> $\frac{19}{10} = 1.9$

= 1 (integer)

(19 - 1x10 = 9)

=> $\frac{9}{5} = 1.8$

= 1 (integer)

(9 - 1x5 = 4)

=> $\frac{4}{2} = 2$

(4 - 2x2 = 0)

```
#include<stdio.h>
int main()
{
    int amount, times;
    Printf("Enter amount ");
    Scanf("%d", &amount);
    If (amount >= 5000)
{
    times = amount/5000;
    amount = amount - times*5000;
    Printf("5000 = %d", times); }
    If (amount >= 1000)
{
    times = amount/1000;
    amount = amount - times*1000;
    Printf("1000 = %d", times); }
    If (amount >= 500)
{
    ——
    ——
    ——
}
```

30287

287

else if

```
If (amount >= 100)
{
    times = amount/100;
    amount = amount - times * 100;
    Printf("100 = %d", times); }
```

87

```
If (amount >=50)
{   times = amount/50;
    amount = amount = times * 50;
    Printf("50 = %d", times); }
```

37

```
If (amount >= 20)
{
   ——————
   ——————
   ——————  }
```

17

```
If (amount >= 10)
{
   ========  }
```

7

```
If (amount >= 5)
{
   ========  }
```

2

```
If (amount >= 2)
```

Factorial Program

```
#include <stdio.h>
int main()
{
    int num, f, c=1;
Printf("Enter a number");
Scanf("%d", &num);
while(c <=num)
{
    f=f*c
    c++;
Printf("%d", f);
}
```

Increment / Discrement

Prefix & Postfix

x = c++ ; → this function assign value of c to x ~~without~~ then increment

x = ++c ; → this function assign value of c to x after increment.

Program

```
# include <stdio.h>
int main()
{
    int j=3;
    j= j++;
```

i= ++j

1st increment happen on value of j then the value of j assigned to (i)

i= j++

(1st value of j is assigned then increment happen)

Program Str.

```
#include<stdio.h>
#include<conio.h>
int/void main()
{
int n=2, Sum=0;
while(n<=50){
Sum=Sum+n;
n=n+2;}
Printf("Sum of 2+4+8+...+50 = %d", Sum);
}
```

```
void main()
return 0;
getch();

int main
(no need)
```

✓

Programing Fundamental

Do-while loop

```
#include <stdio.h>
int main()
{
int i=0
do {
Printf("Pak\n");
i=i+1; / i++;
} while (i<5);
}
```

Output

Pak
Pak
Pak
Pak
Pak

Structure

```
do {
(statement)
} while (condition);
```

For - loop

Loop
i) Initialization
ii) condition
iii) Increment/Decrement

Structure

```
for( int i;
     i=10 ; i>0 ; i=i-1 ) {

    Printf("%d", i);

}
```

Output
10
9
:
2
1

Program
Print factorial by do while loop

```
#include<stdio.h>
int main()
{
    int f=1, c=1, num;
    Printf("Enter number");
    Scanf("%d", &num);

    do {
        f = f*c;
        Printf("%d", f);
        c++;
    }
    while((c<=num);
    }
```

5 x 4 x 3 x 2 x 1 = 120

f x c = 1 x 1
= 1

f x c = 2 x 1
= 2

f x c
3 x 1 =

Program

Fibonacci Series (using loop) { VIP long Q }

0 1 1 2 3 5 8 13 21 ...

```
#include<stdio.h>
int main()
{
int a=0, b=1, res, i;
while( i<=20){
//res = a+b;
res = res + b;
i++;
//printf(
```

0+1=1

2=1+1

3=2+1

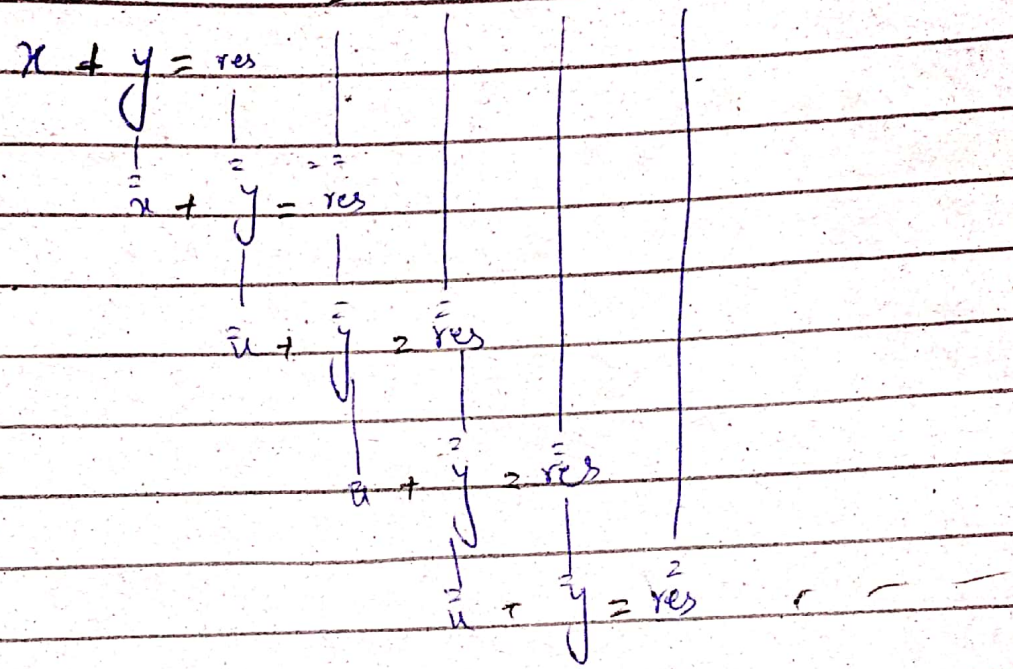
res = 1+1 = 2

res = 2+1

```
{
int x=0, y=1;
int res, i;
while( i<=20 ){
res = x+y;
x = y;
y = res;
Printf("%d", res);
i++;
}
```

Fibonacci Series

0 1 1 2 3 5 8 13 21 ...



Program

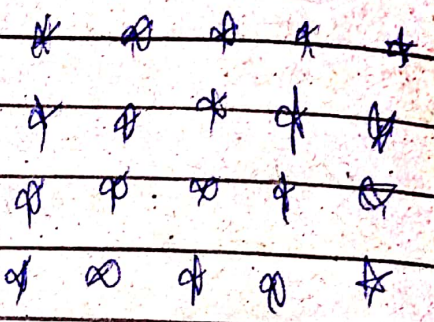
User 01234. --

Print . -- 43210

Program

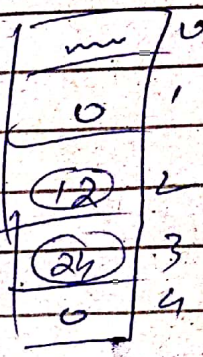
```
#include<stdio.h>
int main()
{
int i,n;
for(i=1; i<5; i++){
    for(n=1; n<5; n++){
        Printf("*");
    }
Printf("*\n");
}
}
```

Print

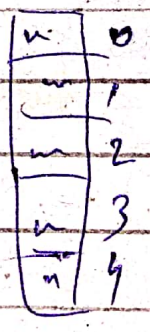


Program

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

int main()
{
    int mark[5] = {0}
    marks[3] = 24;
    mark[2] = 12;
    scanf("%d", &mark[0]);
    printf("%d", mark[0]);
}
```




```
#
#
int main()
{
    int marks[5] = {0};
    for(i=0; i<5; i++)
    {
        scanf("%d", &marks[i]);
        Printf("%d", marks[i]);
    }
```



Program to print max no using function.

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

void max();

int main()
{

max();

max();


}

void max(){
int a,b
Printf("Enter 2 no");
Scanf("%d %d", &a, &b);

If (a>b){
Print("a is greater");}
else
Print("b is greater");
```

Program to swap numbers

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

int swap(int a, int b);
int swap(int a, int b)
{
    int n;
    n = a;
    a = b;
    b = a;
printf("%d", a);
printf("%d", b);
}
int main()
{
    int x, y;
printf("Enter 1st number");
scanf("%d", &x);
printf("Enter 2nd number");
scanf("%d", &y);
swap(x, y);
}
```

Program

to print

```
ααααα
αααα
ααα
αα
α
```

using function

```
#~~
#~~
void printStar( );
int main( )
{
 int x=5
 for( int a=1 ; a<=5; a++ ){

  for (int b=1 ; b<=x ; b++){
   printStar( );
  }
  printf("\n");
  x-- ;
 }

void printStar( )
{
 printf("*");
}
```

①

②

Program

Program to Sum using function

```
#include<stdio.h>
#include<conio.h>
int sum(int a, int b);
int main
{
    int x,y,z;
    x= Sum( 3 , 5 );

    y = Sum( 2,6);

    z= Sum (1 , 3) ;
Printf(" %d\n %d\n %d\n", x,y,z);
}

int sum(int a, int b)
{
    int c;
    c= a+5;
Printf(" %d", c);
OR
return c;
}
```

Output

8
8

34

int n;
a = 34
n = 9;
a = b;

34 = a
a = b
b = 34

Program

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

void swap(int a, int b);

int main()
{ int rem, num, res;
printf("Enter 2 numbers \n");
scanf("%d %d", &num, &b);

void swap(int a, int num);
```

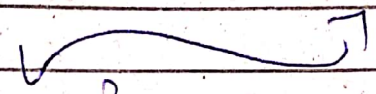
3
2 | 34
30
4

```
}

void swap(int num, int b)
{
rem = num % 10;
res = num / 10;
num = res * 10 + rem;
printf("%d num");
}
```

3
4

=> Local Variable:
Variable having scope within the function

=> Global Variable:
Variable having scope ~~with~~ in the whole program.



Program
to pass two values (2, 3)
base power

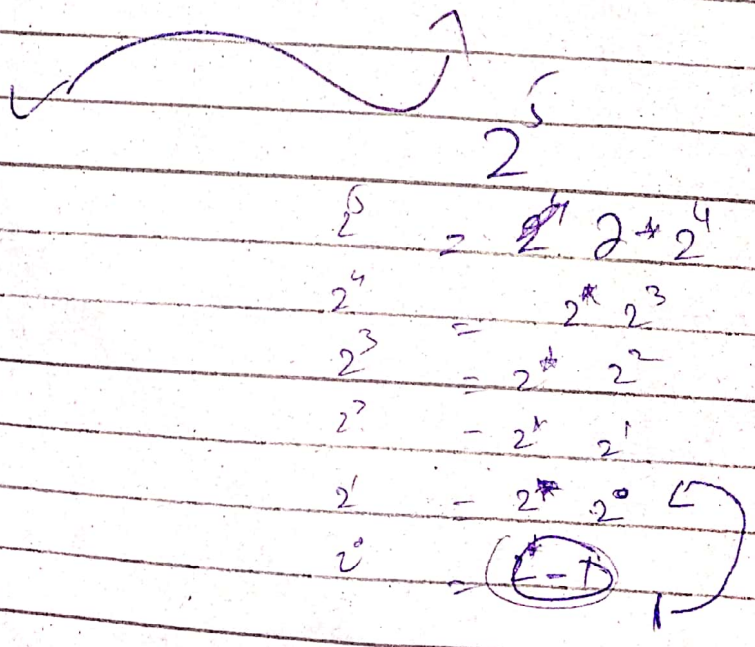
```
#include<stdio.h>
#include<conio.h>

int power(int a, int b);

int main()
{
  int x, y;
  Printf("Enter a number");
  Scanf("%d", &x);
  printf("Enter its power");
  Scanf("%d", &y);

  power(x, y); }
```

2^3

2x2 = 4
4x2 = 6

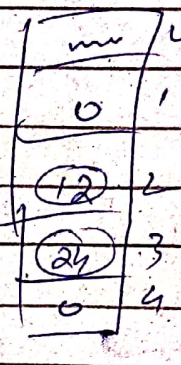
```
int power(int a, int b)
{
    int res, d=1;

    for (int c=1; c<=b; c++){
        res = a*d;
        d = res;
    }
    Printf("%d", res);
}
```

res = a*d
res = 2*1
res = 2
d = 2

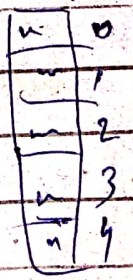


Program

```
#include <stdio.h>
#include <conio.h>
int main()
{
    int mark[5] = {0}
    marks[3] = 24;
    mark[2] = 12;
    scanf("%d", &mark[0]);
    printf("%d", mark[0]);
}
```




```
#
#
int main()
{
    int marks[5] = {0};
for(i=0; i<5; i++)
{
    scanf("%d", &marks[i]);
    Printf("%d", marks[i]);
}
```



Program

To print how many times 24 occurs in given array.

```
#include
#include

int main()
{
int marks[15]={24,..24,..24,...24,..24,..};
int a, b=0
for[a=0; a<=15; a++]
{
if (marks[a]==24)
{ b=b+1; }
}
printf("%d", b);
}
```

Program

To find sum of array marks[15] = { - - }

```
#____
#____
int main()
{
    int sum=0, a;
    int marks[15]={-- - - - };
    for (a=0; a<16; a++)
    {
        sum = sum + marks[a];
    }
    printf("%d", sum);
}
```

Program

① To print sum of numbers ÷ by 4 in array[100]

② To swap numbers of 2 array[5].

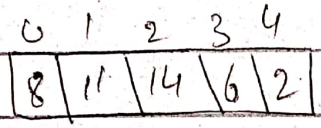
③ Arrays Pass To Function.

23/01/2020 Programing Fundamentals

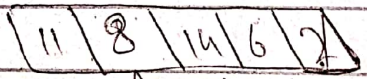
Sorting Arrays

Bubble Sorting

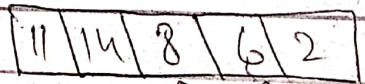
eg①



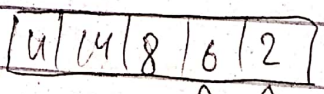
compare & swap



c & s

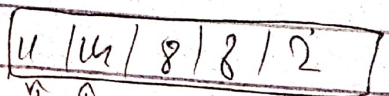


c & s

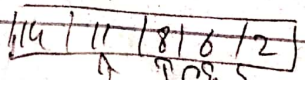


c & s

Again

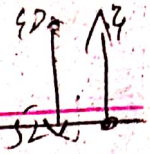


c & s

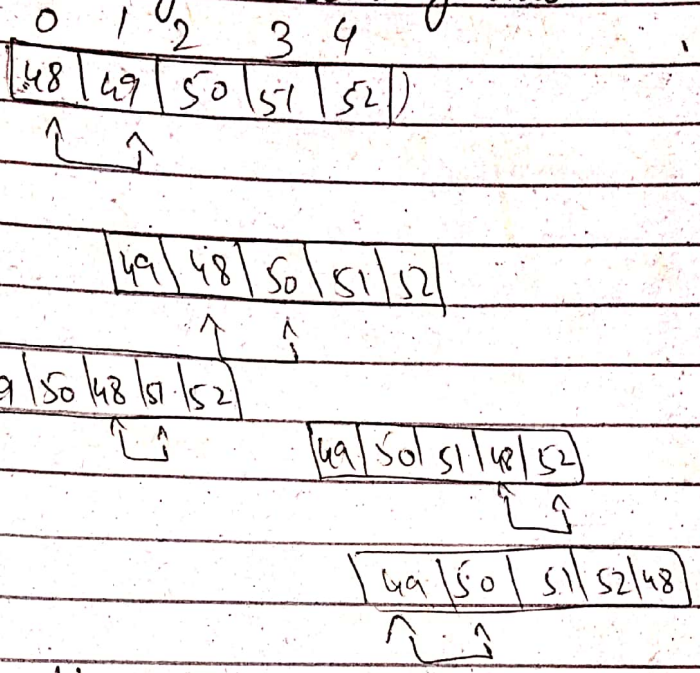


$\sqrt{9-x^2}$

$\sqrt{5^2-x^2}$



e.g (2) descending to descending order



Coding

```
#
#
int main ( )
{
    int arr[5] = {48,49,50,51,52};

for ( int j=0 ; j<4 ; j++ ){
```

~~for (int i=0 ; i<4-j ; i++)~~

```
for (int i=0 ; i<n-j; i++ )
{
    if (arr[i] < arr[i+1]) {
        int temp = arr[i];
        arr[i] = arr[i+1];
        arr[i+1] = temp; }
}
```

Program to print

```
1 2 3 4 5
2 4 6 8
3 6 9
4 8
5
```

```
int main()
{
    for(int a=1; a<=5; a++){①
        for(int b=1; b<a; b++){②
            printf(" ");}②
        for(int c=1; c<=6-a; c++){③
            printf("%d", a*c);
        }③
        printf("\n");
    }①
}
```

=> Program to Search Array

0	1	2	3	4	5	6
8	2	-1	-5	63	0	13

```
int main()
{
    int n;
    int loc = -1;
    printf("Enter no to search");
    scanf("%d", &n);
    for(int i=0; i<6; i++){
    if(arr[i] == n){
        loc = i;
        break;
    }
}
if(loc == -1)
    printf("not found");
else
    printf("%d", loc);
```

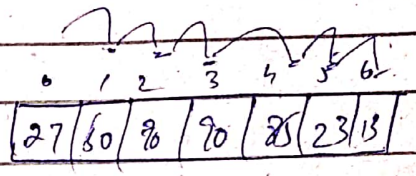
01/02/2020 Programing Fundamental

Searching Array.

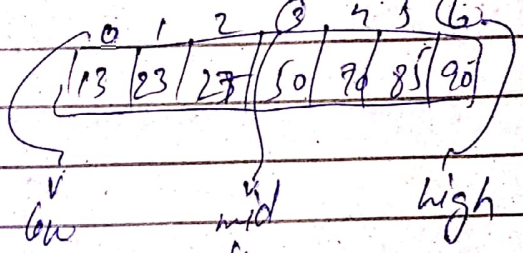
1- Linear Searching ✓

2- Binary Searching

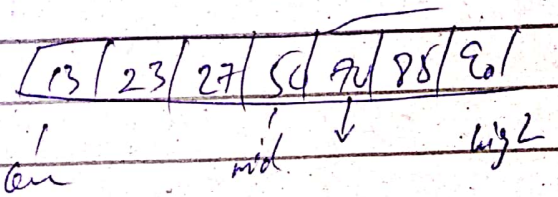
(Binary Searching)

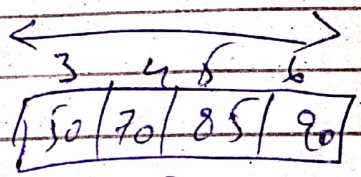


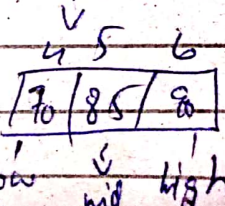
, sort

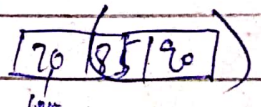


search 70









low = mid + 1 ; 4+6

Program (Binary Searching) VIP

```
#
#
int main ( )
{
  int arr[10];

  int low = 0, high = 9;
  int num;
  printf( );
  scanf ( &num), (-> num = 67;)

  int ind = -1;

  while (low <= high) {

  mid = (low + high)/2;

  If ( arr[mid] = = num ) {

      ind = mid;
      break;

  }
```

```
else if(num > arr[mid])
    low = mid + 1;
}

else {
    high = mid - 1;
}

{ if(ind != -1) {
    printf("No found at %d", ind);
}
else
    printf("No not found");
}
```

2 4 6 8 10
4 16 36 64 100
80 100 100 = 220

= 220

| 23 | 12 | 6 | 27 | 11 | 15 | 0 | 20 | 98 | 74 |

Sum of square numbers

```
#
int main()
{
int array[10];
int sum, y=0

for(int i=0; i<10; i++)
{
printf("Enter number");
scanf("%d", &array[i]); }

for(int n=0 ; n<10 ; n++)
{ If( array[n] % 2==0 ){

sum = array[n] * array[n]

y = x+y;

printf("%d", sum);
```

7 = 7×7 + 0
49 = 7×7 + 0
68 = 3×3 + 4×4
= 4×4 + 6×6
x = 7×7
y = x + y

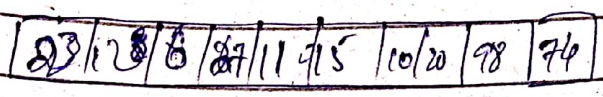
y=0

4=4

Assignment

Selection Sorting Algorithm

Find larger number




```
#include
int main()
{
    int arr[10] = {
for(int n=0; n<10; n++){
    for(int i=0; i<10; i++){

    if(arr[i] > arr[i+1])
    {
        printf("%d", arr[i]);
        n = arr[i];
    }
    else
    {
        n = arr[i+1];
    }
```

#
#

```
int main( )
{
int arr

for(int j=0 ; j<4 ; j++ ){
        int small
for( int i= 1; i<5 ; i++){
if (arr[i] < arr[small]){
        small = i;
}
}
        if (small != i) {
int temp = arr [small];
arr [small] = arr[ i ];
arr[ i ] = temp; }
}
```

Multidimensional Array

Factorial of 5

① By loop

② By recursion

$5! = 4! * 5$

$4! = 3! * 4$

$3! = 2! \times 3$

$2! = 1! \times 2$

$1! = 0! \times 1 \quad \because 0! = 1$

factorial(5);

factorial(4) * 5;

⋮

Q. Write a program to find 2^3

$2^3 = 2 \times 2 \times 2$

$2^3 = 2^2 \times 2$

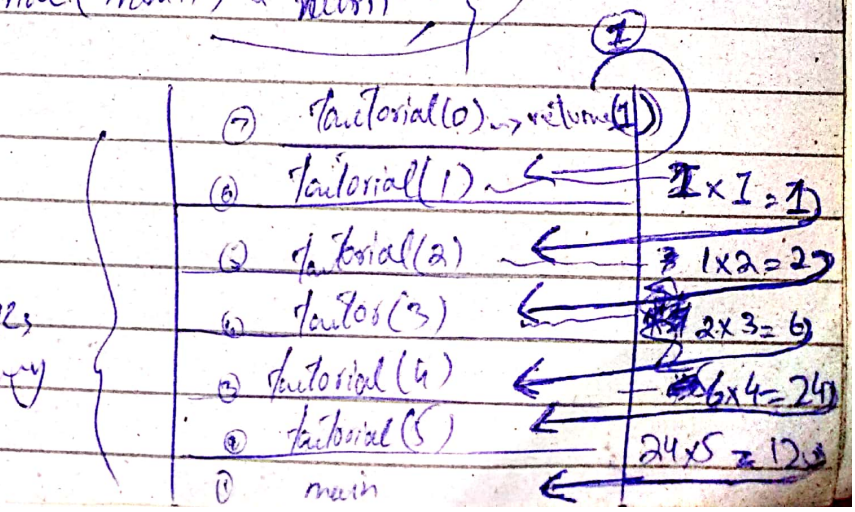
$2^2 = 2^1 \times 2$

$2^1 = 2^0 \times 2 \quad \because 2^0 = 1$

Ficlorial by Recurssion Method

Coding — Assignment of 2^4

```
#
#
int main( )
{
    int num;
    Printf("Enter number for Factorial");
    Scanf("%d", & num );
    int res = factorial( num);
    printf("%d", res );
}

int factorial (int num)
{
    if (num==0)
    { return 1; }
else
{ return factorial(num-1) * num ; }
}
```



Grammer English English

Fundamental Programming Calculating

Power using recursion method

Coding

```
int main()
{
    int n, p;
    Printf("Enter number & Power");
    Scanf("%d\n%d", &n, &p);
    int res = power(n, p);
    printf("%d", res);
}
int power(int n, int p)
{
    if(p==0)
    {return 1;}
    else
{return power(n, p-1) * n;}
}
```

Power using recursion method.

$2^3 = 2 \times 2 \times 2$

$2^3 = 2^2 \times 2$

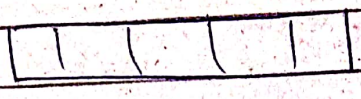
$2^2 = 2^1 \times 2$

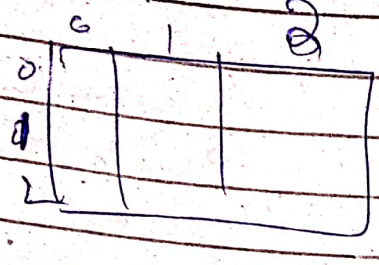
$2^1 = 2^0 \times 2$

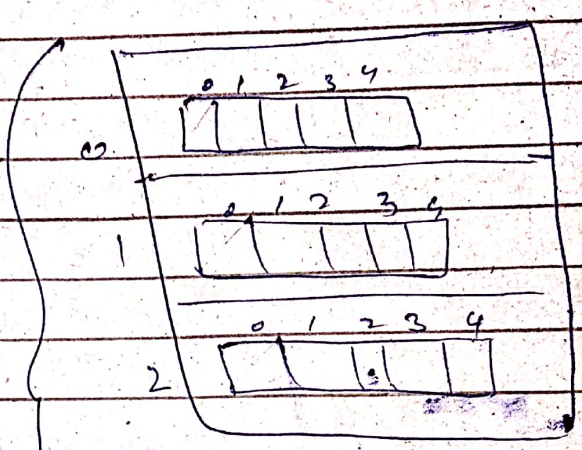
14/02/2020

2D-Arrays

(1D array)

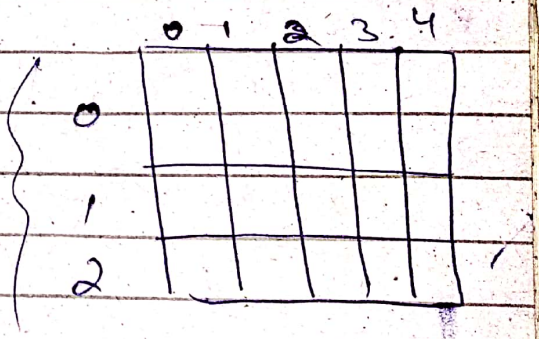






2D array

OR



2-D array

Declaration

int arr[3][5]

Accessing method

arr[row][column]

columns 5

rows 3

= 3x5

Declare value

int arr[3][5] = {{-,-,-,-,-},{-,-,-,-,-},{-,-,-,-,-}}

Assignment

2 2D array & print their sum

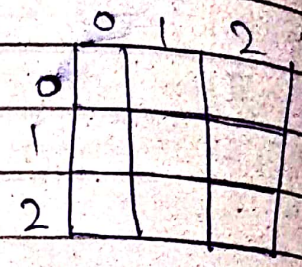
(rows x colum) [ ] + [ ] = [ ]

(rows, colum ↓ User)

Program

To sum two 2D arrays
rows & column given by users

```
#include<stdio.h>
int main()
{
int arr1[row][colum];
int arr2[row][colum];
int arr3[row][colum];

Printf("Enter number of arr1");
Scanf("%d", arr1[          parallel Extra

Printf("Enter no of rows");
Scanf("%d", &row);
Printf("Enter no of columns");
Scanf("%d", &column);

Printf("Enter elements of 1st array");

for(int i=0; i<row; i++)
{   for(int k=0; k<column; k++)
    { scanf("%d", &arr1[i][k]); }
}
```



```
printf("Enter Elements of 2nd array");
for (int x=0; x<row; x++)
{
    for (int y=0; y<column; y++)
    { scanf("%d", arr2[x][k]); }
}
    for (int c=0; c<row; c++)
{
for (int d=0; d<column; d++)
{
}
arr3[c][d] = arr1[c][d] + arr2[c][d];
printf("%d", arr3[c][d]);
}
}
```

3 3
1 2 0 1 2

	0	1	2
0	1	3	4
1	2	3	4
2	2	3	4

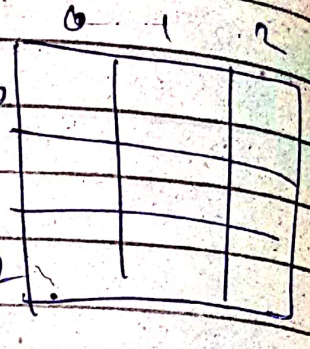
	0	1	2
0	2	3	4
1	5	6	7
2	9	9	9

	0	1	2
0	18	12	13
1	22	12	13
2	22	12	13

Program

Same

```
#include<stdio.h>
int main()
{
    int row, column;
    int arr1[row][colum];
    int arr2[row][column];
    int arr3[row][column]

Printf("Enter no of rows\n");
Scanf("%d", &row);
Printf("Enter no of column\n");
Scanf("%d\n"; &column);


for(int a=0; a<row; a++)
{
    for(int b=0; b<column; b++)
    {
        Printf("Enter element of arr1[%d][%d]", a, b);
        Scanf("%d\n", &arr1[a][b]);
    }
}
```



```
for(int i=0; j<row; i++)
{
    for(int j=0; j<column; j++)
    {
        Printf("Enter element of arr2[%d][%d]\n",i,j);
        Scanf("%d", &arr2[i][j]);
    }
}

for(int x=0; x<~~column~~ row; x++)
{
    for (int y=0; y<column; y++)
    {
        arr3[x][y]= (arr1[x][y])+(arr2[x][y])
        Printf("%d\t", arr3[x][y]);
    }
    Printf("\n");
}
```

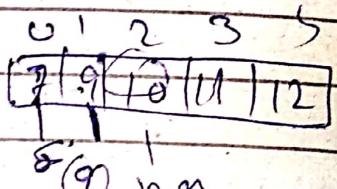
ZAIN

Program Assignment

$$\begin{bmatrix} 2 & 3 \\ 5 & 4 \end{bmatrix} \begin{bmatrix} 1 & 5 \\ 3 & 9 \end{bmatrix} = \begin{bmatrix} 2+9 & 10+27 \\ 5+12 & 25+36 \end{bmatrix}$$

Program to print max no user/direct

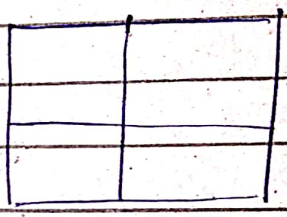
```
#include<stdio.h>
int main()
{
    int max = arr[0];
    int arr[10] = { . . . . . };

    for(int i=0 ; i<10; i++)
    {
        If (max < arr[i])
        { max = arr[i] }
    }
    printf("%d", max);
```



Program

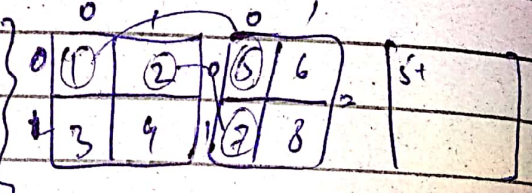
$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix} = \begin{bmatrix} 5+14 & 6+16 \\ 15+28 & 18+32 \end{bmatrix}$$

```
#include<stdio.h>
int main()
{
int rows,col,i,j;
Printf("Enter no of rows & columns\n");
Scanf("%d %d",& row, & col);

int arr1[row][col];
Int arr2[row][col];
int arr3[row][col];

for(i=0; i<row ; i++)
{
    for(j=0; j<col; j++)
    {
        printf("Enter no on arr1[%d][%d]", i,j);
        scanf("%d\n", & arr1[i][j]);
    }

    printf("\n");
}
```



```
for(i=0; i<row; i++)
{
    for(j=0; j<col; j++)
    {
        printf("Enter in arr2[%d][%d]\n", i, j);
        Scanf("%d\n", &arr2[i][j]);
    }
    printf("\n");
}
```

```
for(i=0; i<row; i++)
{
    j=0;
    while(j<2)
    {
```



1x5 + 2x7 \ 1x6+2x8

String : sum of characters

Chapter 4 & 8 Array: Continuous data allocation that stores data of same type.

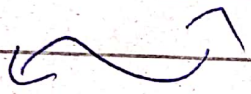
~~Control Structure~~

22/02/2020

String concadination

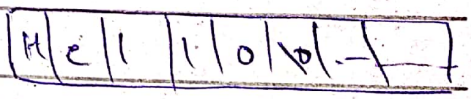
#

```
int main()
{
char str1[ ] = "Hello";
char str2[ ] = "world";
char str3[ ];
```

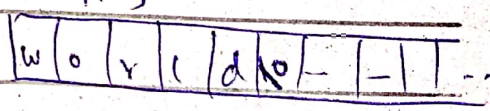


String concadination.

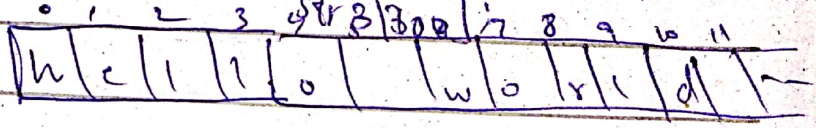
str1[100] = "Hello";



str2[100]



str3[300];



```
#include<stdio.h>
int main()
{
    char str1[100]="Hello";
    char str2[100]="world";
    char str3[300];

    int i=0;

for(int i=0; str1[i]!='\0'; i++){

    str3[i]=str1[i];

}

    str3[i]=' ';
      i++

for(int j=0; str2[j]!='\0'; j++){

    str3[i]=str2[j];
      i++;

}
```

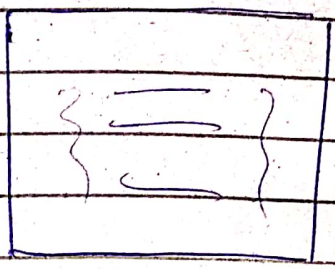
or

Functions

str cpy(,);
str cat(,);
strlen( );

File Handling

file.txt



① Create / Open

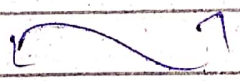
② read / write

③ close

address : xxyy31

to store address of file used pointer

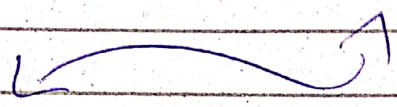
FILE *ptr



#

```
int main()
{
    FILE *ptr ; char name[20];
    ptr = fopen("E://Test.txt", "w");
    if (ptr == NULL)
    { printf("File not created"); }
```

```
else
{
    scanf("%s", name);
    fprintf(ptr, "%s", name);
    fclose(ptr);
}
}
```



```
    else
    {
        scanf(ptr, "%s", name);
        printf("%s", name);
    }
```

#
#

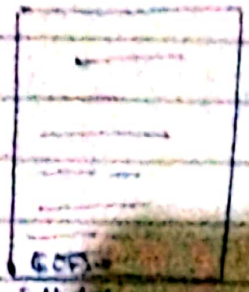
```
int main() {

FILE *ptr; char name[20]; int roll, marks;

ptr = fopen("F://test.txt", "w");
If (ptr== NULL)
{ printf("                ");
}
else
{
scanf("%s %d %d", &name, &roll, &marks);
fprintf(ptr "%s \n %d \n %d \n",
                name, roll no, marks);
fclose (ptr);
```

(5 Student records (using loop))

```
#
#
int main()
{

char str[100]; char x = 'x';
    //open
    FILE * fptr;

gets(str);
  // scan value/line from users
     and save it into str

fputs(str, fptr);

    // prints str value in location fptr of file
       print complet line

fgets(str,50, fptr);

    // scan complet line
    // fptr -> file address
    & store file content in str
     50 characters from file including spaces

fgetc(x, fptr)

    // reads 1 characters from
       file
```

fputc (K , ptr)

// print one character in file




```
#include <stdio.h>
#
int main()
{
    FILE *ptr ;// Declaration of pointer

ptr = fopen("E://test.txt", "r");
```

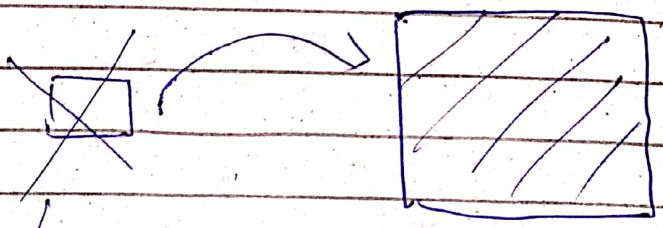
```
fgetc(ch, ptr);
while (ch != EOF)
{
    printf("%c", ch);
    fgetc(ch, ptr);
}
    fclose(ptr1);
```

To print all characters of one file into other

```
while (ch != EOF)
{
    printf("%c", ch);
    fgetc(ch, ptr);
    fputs(ch, ptr2);
}
```

File
loops
arrays
str
functions

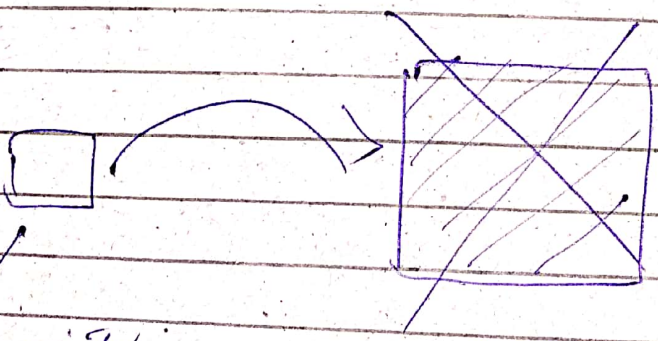
Dangling Memory



Deleted

memory
Dangling Memory

Dangling Pointer



Dangling pointer

memory

Deleting using
function

Function Overloading

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

(making same sign - function with diff parameters)

=> Function overloading means using the same name function with different parameters

Function overriding means using the same name function with different implementation.