

24 Friday



February 2017

February 2017



Saturday 25

CHAPTER - 5

Week 8

Functions and recursion

08:30

Function :-

Block of code that performs particular task.

09:00

Take → [Do work] → return result.

10:00

* in can be used multiple times to increase code readability, reusability.

01:00

3 step process in C

02:00

(i) Function Prototype / Declaration :- > Tell the compiler

04:00

void printHello();
function-name

05:00

(ii) Function Definition :- > Do the work

06:00

void printHello() {
printf("Hello");
}

07:00

(iii) Function call :-

int main () {

 printHello();

↓

use the work

February						
S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28

March						
S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28

Simple example

05:30

```
#include <stdio.h>
void printHello();
int main()
{
    printHello();
}
```

10:00

12:00

→ Functions are very useful when you write code in company.

01:00

Practice Qs 27

02:00

④ Write 2 functions - One to print "Hello" and second to print "good bye".

03:00

04:00 Practice Qs 28

④ Write a function that prints "Namaste" if user is Indian & "Bonjour" if the user is French.

Sunday 26

27 Monday



February 2017

Week 9

059-307

Properties :-

- Execution always start from main function.
- A Function gets called directly or indirectly from main
- There can be multiple function in a program.

Function Types

- 12.00 ✓ **④ "Library functions"** → declared & defined by the programming language
- 01.00 → special functions
inbuilt inc

Ex:- `scanf()`, `printf()` etc. etc. and → built-in functions like `sqrt()`, `pow()`, `random()` etc.

Function Types

- 04.00 **④ "User-defined functions"** → declared & defined by the programmer
- 01.00 → void printHello();

Ex:- `scanf()`, `printf()` etc. etc. and → built-in functions like `sqrt()`, `pow()`, `random()` etc.

04.00 **Passing Arguments**
Programs, which are written in C language, store their variables in memory. Functions can take values of arguments as parameters. These values are passed to the function as arguments.

04.00 Functions can take values of arguments as parameters. These values are passed to the function as arguments.

04.00 `int sum(int a, int b);`

04.00 `int s = sum(a, b);`

04.00 `int sum(int a, int b){`

04.00 `int s = a + b;`

04.00 `return a+b;`

04.00 }
04.00 }
04.00 }
04.00 }
04.00 }
04.00 }

04.00 Program :- When we have to calculate sum of two numbers, we write function to print table of a number (`n`) given by user.

04.00 `void printTable(int n);`

04.00 `printTable(n);`

04.00 `int sum(int a, int b);`

2017 February
S M T W T F S S M T W T F S
1 2 3 4 5 6 7 8 9 10 11
12 13 14 15 16 17 18 19 20 21 22 23 24 25
26 27 28 29 30 31

28 Tuesday



Week 9

059-306

- Program for taking two user integers as input and displaying write a function to get a sum of them.

```
09.00 #include <stdio.h>
09.00 int sum( int a, int b );
09.00 int main()
09.00 {
09.00     int a, b ;
09.00     printf("Enter a and b: ");
09.00     scanf("%d %d", &a, &b);
09.00     int s = sum(a, b);
09.00     printf("\n%d\n", s);
09.00 }
```

```
09.00 #include <stdio.h>
09.00 void printTable(int n);
09.00 int sum( int a, int b );
09.00 int main()
09.00 {
09.00     int i, k;
09.00     for( i=1; i<=n; i++ )
09.00     {
09.00         for( k=1; k<=i; k++ )
09.00             printf("%d ", k);
09.00         printf("\n");
09.00     }
09.00 }
```

Tuesday 28



Week 9

059-307

03 Friday



March 2017

Recursion...

Week 9

09:00 when a function calls itself, it's called recursion.

10:00 Practice Qs 32

11:00 $\text{Q} \# \text{ print "Hello world" 5 times using recursion}$

#include <stdio.h>

09:00 void printfW (int count);

10:00 void main()

11:00 {

12:00 void printfW (int count) {
if (count == 0) {
return;
}
printf ("Hello world ");
printfW (count - 1);
}

13:00 }

14:00 }

15:00 }

16:00 }

17:00 }

18:00 }

19:00 }

20:00 }

21:00 }

22:00 }

23:00 }

24:00 }

NOTE:- The work which will solve by loops that is also solved by recursion and vice-versa.

but we have to choose the efficient one.

2017	March
S M T W T F S	S M T W T F S
1 2 3 4 5 6 7 8	9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31

March 2017



March 2017

understanding the concept of Recursion by math.

$$f(n) = n^2$$

$$f(1) = 1$$

$$f(2) = 4 \Rightarrow \text{but } f(f(2)) = 16$$

$$f(3) = 9$$

$$f(f(3)) = 81$$

$$f(f(f(3))) = 729$$

$$f(f(f(f(3)))) = 6561$$

$$f(f(f(f(f(3))))) = 59049$$

$$f(f(f(f(f(f(3)))))) = 531441$$

$$f(f(f(f(f(f(f(3))))))) = 4782969$$

$$f(f(f(f(f(f(f(f(3)))))))) = 43046721$$

$$f(f(f(f(f(f(f(f(f(3)))))))) = 390625$$

$$f(f(f(f(f(f(f(f(f(3)))))))) = 359375$$

$$f(f(f(f(f(f(f(f(f(3)))))))) = 32768$$

$$f(f(f(f(f(f(f(f(f(3)))))))) = 295375$$

$$f(f(f(f(f(f(f(f(f(3)))))))) = 262144$$

$$f(f(f(f(f(f(f(f(f(3)))))))) = 2304$$

$$f(f(f(f(f(f(f(f(f(3)))))))) = 2048$$

$$f(f(f(f(f(f(f(f(f(3)))))))) = 17214$$

$$f(f(f(f(f(f(f(f(f(3)))))))) = 147456$$

12:00 Normal Function Call

09:00 main → [function]
02:00 return

13:00 Recursion Function Call

09:00 main → [f(m)] → [f(m)] → [f(m)] → [f(m)]

05:00 return

17:00 return

21:00 return

Sunday 05

Saturday 04



* so recursion functions calls itself for n number of times.

06 Monday



March 2017

March 2017



Tuesday 07

Practice Qs (30) ~~of 30~~ of 30 ~~of 30~~ of 30
Sum of first 'n' natural numbers. by recursion.

$$\begin{aligned} \text{sum}(n) &= n + 1 + 2 + 3 + 4 + \dots + (n-1) + n \\ &\quad \left[\text{sum}(n-1) + n \right] \\ 5 &\rightarrow 1 + 2 + 3 + 4 + 5 \\ 4 &\rightarrow 1 + 2 + 3 + 4 \\ 3 &\rightarrow 1 + 2 + 3 \\ 2 &\rightarrow 1 + 2 \\ 1 &\rightarrow 1 \end{aligned}$$

$$\begin{aligned} \text{sum}(2) &+ 2 \quad \text{or} \quad \text{sum}(n-1) + 2 \\ \text{sum}(3) &+ 3 \quad \text{or} \quad \text{sum}(n-1) + 3 \\ \text{sum}(4) &+ 4 \quad \text{or} \quad \text{sum}(n-1) + 4 \\ \text{sum}(5) &+ 5 \quad \text{or} \quad \text{sum}(n-1) + 5 \end{aligned}$$

$$\begin{aligned} \text{sum}(1) &+ 1 \\ 1 &\rightarrow 1 \end{aligned}$$

$$\begin{aligned} \text{sum}(2) &+ 3 \\ &\quad \downarrow \\ &\quad \text{sum}(1) + 2 = 1 + 2 = 3 \end{aligned}$$

$$\begin{aligned} \text{sum}(3) &+ 4 \\ &\quad \downarrow \\ &\quad \text{sum}(2) + 3 = 3 + 4 = 7 \end{aligned}$$

$$\begin{aligned} \text{sum}(4) &+ 5 \\ &\quad \downarrow \\ &\quad \text{sum}(3) + 4 = 7 + 4 = 11 \end{aligned}$$

$$\begin{aligned} \text{sum}(5) &+ 6 \\ &\quad \downarrow \\ &\quad \text{sum}(4) + 5 = 11 + 5 = 16 \end{aligned}$$

$$\begin{aligned} \text{sum}(6) &+ 7 \\ &\quad \downarrow \\ &\quad \text{sum}(5) + 6 = 16 + 6 = 22 \end{aligned}$$

$$\begin{aligned} \text{sum}(7) &+ 8 \\ &\quad \downarrow \\ &\quad \text{sum}(6) + 7 = 22 + 7 = 29 \end{aligned}$$

$$\begin{aligned} \text{sum}(8) &+ 9 \\ &\quad \downarrow \\ &\quad \text{sum}(7) + 8 = 29 + 8 = 37 \end{aligned}$$

$$\begin{aligned} \text{sum}(9) &+ 10 \\ &\quad \downarrow \\ &\quad \text{sum}(8) + 9 = 37 + 9 = 46 \end{aligned}$$

$$\begin{aligned} \text{sum}(10) &+ 11 \\ &\quad \downarrow \\ &\quad \text{sum}(9) + 10 = 46 + 10 = 56 \end{aligned}$$

$$\begin{aligned} \text{sum}(11) &+ 12 \\ &\quad \downarrow \\ &\quad \text{sum}(10) + 11 = 56 + 11 = 67 \end{aligned}$$

$$\begin{aligned} \text{sum}(12) &+ 13 \\ &\quad \downarrow \\ &\quad \text{sum}(11) + 12 = 67 + 12 = 79 \end{aligned}$$

$$\begin{aligned} \text{sum}(13) &+ 14 \\ &\quad \downarrow \\ &\quad \text{sum}(12) + 13 = 79 + 13 = 92 \end{aligned}$$

$$\begin{aligned} \text{sum}(14) &+ 15 \\ &\quad \downarrow \\ &\quad \text{sum}(13) + 14 = 92 + 14 = 106 \end{aligned}$$

$$\begin{aligned} \text{sum}(15) &+ 16 \\ &\quad \downarrow \\ &\quad \text{sum}(14) + 15 = 106 + 15 = 121 \end{aligned}$$

$$\begin{aligned} \text{sum}(16) &+ 17 \\ &\quad \downarrow \\ &\quad \text{sum}(15) + 16 = 121 + 16 = 137 \end{aligned}$$

$$\begin{aligned} \text{sum}(17) &+ 18 \\ &\quad \downarrow \\ &\quad \text{sum}(16) + 17 = 137 + 17 = 154 \end{aligned}$$

$$\begin{aligned} \text{sum}(18) &+ 19 \\ &\quad \downarrow \\ &\quad \text{sum}(17) + 18 = 154 + 18 = 172 \end{aligned}$$

$$\begin{aligned} \text{sum}(19) &+ 20 \\ &\quad \downarrow \\ &\quad \text{sum}(18) + 19 = 172 + 19 = 191 \end{aligned}$$

$$\begin{aligned} \text{sum}(20) &+ 21 \\ &\quad \downarrow \\ &\quad \text{sum}(19) + 20 = 191 + 20 = 211 \end{aligned}$$

$$\begin{aligned} \text{sum}(21) &+ 22 \\ &\quad \downarrow \\ &\quad \text{sum}(20) + 21 = 211 + 21 = 232 \end{aligned}$$

$$\begin{aligned} \text{sum}(22) &+ 23 \\ &\quad \downarrow \\ &\quad \text{sum}(21) + 22 = 232 + 22 = 254 \end{aligned}$$

$$\begin{aligned} \text{sum}(23) &+ 24 \\ &\quad \downarrow \\ &\quad \text{sum}(22) + 23 = 254 + 23 = 277 \end{aligned}$$

$$\begin{aligned} \text{sum}(24) &+ 25 \\ &\quad \downarrow \\ &\quad \text{sum}(23) + 24 = 277 + 24 = 301 \end{aligned}$$

$$\begin{aligned} \text{sum}(25) &+ 26 \\ &\quad \downarrow \\ &\quad \text{sum}(24) + 25 = 301 + 25 = 326 \end{aligned}$$

$$\begin{aligned} \text{sum}(26) &+ 27 \\ &\quad \downarrow \\ &\quad \text{sum}(25) + 26 = 326 + 26 = 352 \end{aligned}$$

$$\begin{aligned} \text{sum}(27) &+ 28 \\ &\quad \downarrow \\ &\quad \text{sum}(26) + 27 = 352 + 27 = 379 \end{aligned}$$

$$\begin{aligned} \text{sum}(28) &+ 29 \\ &\quad \downarrow \\ &\quad \text{sum}(27) + 28 = 379 + 28 = 407 \end{aligned}$$

$$\begin{aligned} \text{sum}(29) &+ 30 \\ &\quad \downarrow \\ &\quad \text{sum}(28) + 29 = 407 + 29 = 436 \end{aligned}$$

$$\begin{aligned} \text{sum}(30) &+ 31 \\ &\quad \downarrow \\ &\quad \text{sum}(29) + 30 = 436 + 30 = 466 \end{aligned}$$

Q) Sum of first 'n' natural numbers. by recursion.

TRACTING
we call sum(5)? 15

sum(4) + 5 = 10 + 5 = 15

sum(3) + 4 = 6 + 4 = 10

sum(2) + 3 = 3 + 3 = 6

sum(1) + 1 = 1 + 1 = 2

Base Case → final case → Answer fix

```

00:00 // recursive function
01:00 int sum (int n) {
02:00     if (n==1) {
03:00         return 1;
04:00     }
05:00     int sumN01 = sum (n-1);
06:00     int sumN = sumN01 + n;
07:00
08:00     return sumN;
}

```

2017	March
S M T W T F S	S M T W T F S
1 2 3 4 5 6 7	8 9 10 11 12 13 14
15 16 17 18 19 20 21	22 23 24 25
26 27 28 29 30 31	

2017	April
S M T W T F S	S M T W T F S
1 2 3 4 5 6 7	8 9 10 11 12 13 14
15 16 17 18 19 20 21	22 23 24 25
26 27 28 29 30 31	

Properties of Recursion

- a. Anything that can be done with iteration, can be done with recursion and vice-versa.

- b. Recursion can sometimes give the most simple solution.

- c. Base Case is the condition which stops recursion.

- d. Iteration has infinite loop & Recursion has Stack overflow.

Q3

- Practice Q3

Q4

Q5

Q6

Q7

Q8

Q9

Q10

Q11

Q12

Q13

Q14

Q15

Q16

Q17

Q18

Q19

Q20

Q21

Q22

Q23

Q24

Q25

Q26

Q27

Q28

Q29

Q30

Q31

Q32

Q33

Q34

Q35

Q36

Q37

Q38

Q39

Q40

Q41

Q42

Q43

Q44

Q45

Q46

Q47

Q48

Q49

Q50

Q51

Q52

Q53

Q54

Q55

Q56

Q57

Q58

Q59

Q60

Q61

Q62

Q63

Q64

Q65

Q66

Q67

Q68

Q69

Q70

Q71

Q72

Q73

Q74

Q75

Q76

Q77

Q78

Q79

Q80

Q81

Q82

Q83

Q84

Q85

Q86

Q87

Q88

Q89

Q90

Q91

Q92

Q93

Q94

Q95

Q96

Q97

Q98

Q99

Q100

Q101

Q102

Q103

Q104

Q105

Q106

Q107

Q108

Q109

Q110

Q111

Q112

Q113

Q114

Q115

Q116

Q117

Q118

Q119

Q120

Q121

Q122

Q123

Q124

Q125

Q126

Q127

Q128

Q129

Q130

Q131

Q132

Q133

Q134

Q135

Q136

Q137

Q138

Q139

Q140

Q141

Q142

Q143

Q144

Q145

Q146

Q147

Q148

Q149

Q150

Q151

Q152

Q153

Q154

Q155

Q156

Q157

Q158

Q159

Q160

Q161

Q162

Q163

Q164

Q165

Q166

Q167

Q168

Q169

Q170

Q171

Q172

Q173

Q174

Q175

Q176

Q177

Q178

Q179

Q180

Q181

Q182

Q183

Q184

Q185

Q186

Q187

Q188

Q189

Q190

Q191

Q192

Q193

Q194

Q195

Q196

Q197

Q198

Q199

Q200

Q201

Q202

Q203

Q204

Q205

Q206

Q207

Q208

Q209

Q210

Q211

Q212

Q213

Q214

Q215

Q216

Q217

Q218

Q219

Q220

Q221

Q222

Q223

Q224

Q225

Q226

Q227

Q228

Q229

Q230

Q231

Q232

Q233

Q234

Q235

Q236

Q237

Q238

Q239

Q240

Q241

Q242

Q243

Q244

Q245

Q246

Q247

Q248

Q249

Q250

Q251

Q252

Q253

Q254

Q255

Q256

Q257

Q258

Q259

Q260

Q261

Q262

Q263

Q264

Q265

Q266

Q267

Q268

Q269

Q270

Q271

Q272

Q273

Q274

Q275

Q276

Q277

Q278

Q279

Q280

Q281

Q282

Q283

Q284

Q285

Q286

Q287

Q288

Q289

Q290

Q291

Q292

Q293

Q294

Q295

Q296

Q297

Q298

Q299

Q300

Q311

Q312

Q313

Q314

Q315

Q316

Q317

Q318

Q319

Q320

Q321

Q322

Q323

Q324

Q325

Q326

Q327

Q328

Q329

Q330

Q331

Q332

Q333

Q334

Q335

Q336

Q337

Q338

Q339

Q340

Q341

Q342

Q343

Q344

Q345

Q346

Q347

Q348

Q349

Q350

Q351

Q352

Q353

Q354

Q355

Q356

Q357

Q358

Q359

15 Wednesday



March 2017

March 2017



Thursday 16



Homework ~~set~~ write a function to find sum of digits of a number
 Q) write a function to find square root of a number

Method 1

Method 2

```

15) #include <stdio.h>
int Dsum(int n);
int main()
{
    int n;
    printf("Enter a number: ");
    scanf("%d", &n);
    int sum = Dsum(n);
    printf("Sum of digits : %d", sum);
}

int Dsum(int n)
{
    int sum=0;
    while (n!=0)
    {
        sum+=sum+(n%10);
        n=n/10;
    }
    return sum;
}

16) #include <math.h>
#include <stdio.h>
void main()
{
    int b;
    printf("Enter n: ");
    scanf("%d", &b);
    float sgroot = sqrt(b);
    printf("Square root of %d is: %f", b, sgroot);
}

int squareRoot(int n)
{
    void main()
    {
        int n=65; // n - user input
        float sgroot = squareRoot(n);
        printf("Square root of %d is: %f", n, sgroot);
    }
}

int squareRoot(int n)
{
    return pow(n, (0.5));
} // return sgroot(n);

```

2017	March
S M T W T F S	S M T W T F S
1 2 3 4 5 6 7	8 9 10 11
12 13 14 15 16 17 18	19 20 21 22 23 24 25
26 27 28 29 30 31	

17 Friday



March 2017

- ③ write a function to print "hot" or "cold" depend upon the temp user enters.

make your own function. (After 3 hours effort than I have found the answer)

$\text{if} (\text{temp} > 12) \{\}$

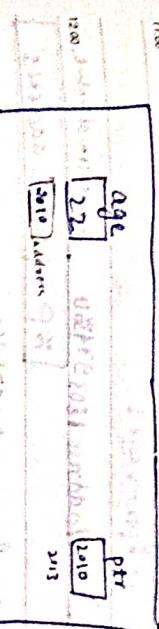
return "hot";

$\text{else if} (\text{temp} \leq 0) \{\}$

return "cold";

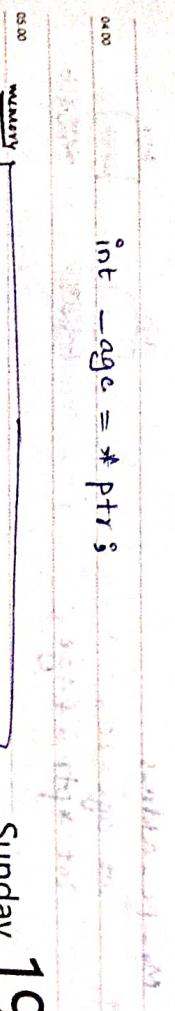
```
else {
    for (int i = 1; i <= power; i++) {
        value = value * number;
    }
    return value;
}
```

"Answer"



syntax: `int *ptr = &age;` → value at address operator (*)

Int *ptr = &age; → address of (&)



Sunday 19

```
int age = 20;
int *ptr = &age;
printf ("%u", *ptr);
printf ("%u", &age);
```

March 2017						
S	M	T	W	F	S	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

April 2017						
S	M	T	W	F	S	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

March 2017



Saturday 18

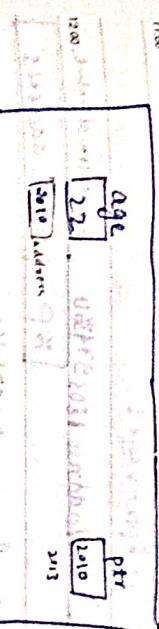
CHAPTER - 6

"Pointers"

*

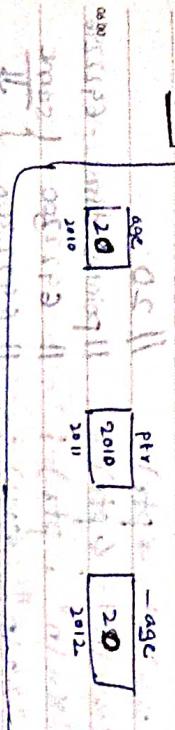
A Variable that stores the memory address of another Variable.

↳ Address



syntax: `int *ptr = &age;` → value at address operator (*)

Int *ptr = &age; → address of (&)



20 Monday



March 2017

Declaring Pointers.

```
int age=20; int *ptr=&age;
char star='*'; char *ptr=&star;
float pi=3.14; float *ptr=&pi;
```

Format specifier: `int age=20;`
`int *ptr=&age;`

```
printf("%p", &age); //Address 1803974360 //hexadecimal value
%P - Pointer address
```

```
printf("%p", ptr); //Address 1803974360
```

u.u - unsigned

```
printf("%p", &ptr); //Address of ptr 1803974358
```

00:00

M_y - PC - address

```
int age=20; [20] [2000] [ptr]
int *ptr=&age; 6922200 6922216
```

00:00

```
printf("%d\n", *ptr); // 20
```

00:00

```
printf("%u\n", *ptr); //Pointers address - 6922200
```

```
printf("%u\n", ptr); // 6922200 {same}
```

00:00

```
printf("%u\n", age); // 6922200
```

*ptr=age

2017	March
S M T W T F S	S M T W T F S
1 2 3 4 5 6 7	8 9 10 11 12 13 14
9 10 11 12 13 14 15	16 17 18 19 20 21 22
16 17 18 19 20 21 22	23 24 25 26 27 28 29
25 26 27 28 29 30	31

March 2017

printf("%d", age); //20

printf("%d", *ptr); //20

printf("%d", &ptr); //20

Tuesday 21



V-GUARD

March 2017

printf("%d", age); //20

printf("%d", *ptr); //20

printf("%d", &ptr); //20

Practice Q1: Find the value of `x` using `ptr`

0	10
---	----

`int x; int *ptr=&x; printf("%d", *ptr);`

00:00

`ptr = &x; printf("%d", *ptr);`

`*ptr = 0; printf("%d", *ptr);`

`*ptr = 0; printf("%d", *ptr);`

```
00:00 printf("%d", *ptr); //0
00:00 printf("%d", &ptr); //0
```

00:00

`*ptr = *ptr + 5; // *ptr += 5; 0+5=5`

00:00

`printf("%d", *ptr); // 5`

00:00

`(*ptr)++; // *ptr = *ptr + 1; 5+1=6`

00:00

`printf("%d", *ptr); // 6`

00:00

`(*ptr)++; // *ptr = *ptr + 1; 6+1=7`

00:00

`printf("%d", *ptr); // 7`

00:00

`(*ptr)++; // *ptr = *ptr + 1; 7+1=8`

00:00

`printf("%d", *ptr); // 8`

00:00

`(*ptr)++; // *ptr = *ptr + 1; 8+1=9`

00:00

`printf("%d", *ptr); // 9`

00:00

`(*ptr)++; // *ptr = *ptr + 1; 9+1=10`

00:00

`printf("%d", *ptr); // 10`

00:00

`(*ptr)++; // *ptr = *ptr + 1; 10+1=11`

00:00

`printf("%d", *ptr); // 11`

00:00

`(*ptr)++; // *ptr = *ptr + 1; 11+1=12`

00:00

`printf("%d", *ptr); // 12`

00:00

`(*ptr)++; // *ptr = *ptr + 1; 12+1=13`

00:00

`printf("%d", *ptr); // 13`

00:00

`(*ptr)++; // *ptr = *ptr + 1; 13+1=14`

00:00

`printf("%d", *ptr); // 14`

00:00

00:00

22 Wednesday



March 2017

Week 12
Program :- SEE EXAM Question

$$a = 6356748$$

$$\&ptr = 6356740$$

```
main()
{
    int a, b;
    int *p;
    a = 15;
    p = &a;
    b = *p;
    printf ("%d and %u\n", a, &a);
    printf ("%d and %u\n", *p, p);
    printf ("%d and %u\n", p, &p);
    printf ("%d and %u\n", b, &b);
    printf ("%d\n", a);
}
```

```
*p = 25;
printf ("%d\n", a);
```

03.00
Pointer

```
04.00  
Memory
```

a	b	ptr
15	15	2010
6356748	6356744	2013
6356740	6356740	2010

```
05.00  
06.00  
07.00  
08.00  
09.00
```

```
01.00  
02.00  
03.00  
04.00  
05.00
```

Syntax
int **pptr;

```
06.00  
07.00  
08.00  
09.00  
10.00
```

```
01.00  
02.00  
03.00  
04.00  
05.00
```

float **ptr;

```
06.00  
07.00  
08.00  
09.00  
10.00
```

```
01.00  
02.00  
03.00  
04.00  
05.00
```

Practice Qs 37

```
06.00  
07.00  
08.00  
09.00  
10.00
```

Print the value of 'i' from its pointer-to-pointer

```
01.00  
02.00  
03.00  
04.00  
05.00
```

eg. int i=5;
int *i=5;
int *ptr = &i;
int **ptr = &ptr;

```
06.00  
07.00  
08.00  
09.00  
10.00
```

print ("%d\n", *(*(ptr))); // **ptr

```
01.00  
02.00  
03.00  
04.00  
05.00
```

2017 March
S M T W T F S S M T W T F S
1 2 3 4 5 6 7 8 9 10 11
12 13 14 15 16 17 18 19 20 21 22 23 24 25
26 27 28 29 30 31

25/

25/

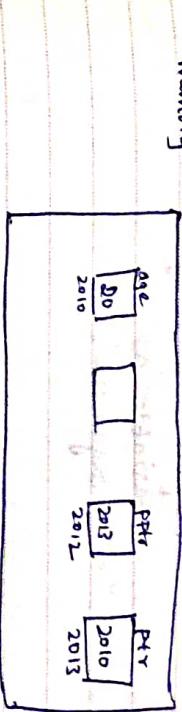
Thursday 23



March 2017

Week 12
Pointer to pointer (For Interviews)

A variable that stores the memory address of another



Week 12
Exam Question

081-244
082-203

081-244
082-203



March 2017

March 2017



Saturday 25

24 Friday

Pointers in function call

Call by reference.

Value

We pass value of variable as argument.

argument

Pass by value

#include <stdio.h>

void square(int n);

void main() {

int number=4;

square(number);

printf ("number = %d\n", number);

}

void square (int n) {

n= n*n;

printf ("square is = %d\n", n);

{

sqrt square is = 16

number = 4

03/03/2017

March

2017

S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

04/03/2017

March

2017

April

S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

Sunday 26

05/03/2017

sqrt square is = 16

number = 4

06/03/2017

March

2017

S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

07/03/2017

March

2017

April

S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

08/03/2017

March

2017

April

S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

09/03/2017

March

2017

April

S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

10/03/2017

March

2017

April

S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

11/03/2017

March

2017

April

S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

12/03/2017

March

2017

April

S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

13/03/2017

March

2017

April

S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

14/03/2017

March

2017

April

S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

15/03/2017

March

2017

April

S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

16/03/2017

March

2017

April

S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

17/03/2017

March

2017

April

S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

18/03/2017

March

2017

April

S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

19/03/2017

March

2017

April

S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

20/03/2017

March

2017

April

S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

21/03/2017

March

2017

April

S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

22/03/2017

March

2017

April

S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

23/03/2017

March

2017

April

S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

24/03/2017

March

2017

April

S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

25/03/2017

March

2017

April

S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12</		

27 Monday



$$\begin{aligned} a &= a^b; \\ b &= a^b; \\ a &= a^b; \end{aligned}$$

March 2017

March 2017



Tuesday 28

Week 13

Practice Qs 38: Swap 2 numbers a & b.

call by value

or
value

08:00 #include <stdio.h>

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:00

08:0

April 2017

03 Monday

Week 14

The V-GUARD logo is located in the bottom right corner of the page. It consists of the word "V-GUARD" in a bold, sans-serif font, enclosed within a black oval border.

April 2017

The V-Guard logo consists of the word "V-GUARD" in a bold, sans-serif font, enclosed within an oval border. To the right of the text is a small, stylized graphic of a person's head and shoulders.

Tuesday 04

~~WEEK 14~~ ~~STEP 7~~

CHARTER

~~SHOP & H~~

Collection at [unclear] [unclear] location.

SYNTHETIC POLY(AMINO ACID AMIDES)

data-type = **Array-name** [**size**] ;

~~AB-EGH~~ int marks [3]; — 1st & 2nd stabs.

```
char name[10];
```

float price [2];

6

卷之三

int \leftarrow $t \rightarrow t \rightarrow t \rightarrow$

6 7 8

卷之三

O-Based indexing

THE JOURNAL OF CLIMATE

```
int market = $97.95; //an
```

April						
		2017				
		S	M	T	W	F
		1	2	3	4	5
		6	7	8	9	10
		11	12	13	14	15
		16	17	18	19	20
		21	22			

2017							April						
S	M	T	W	T	F	S	S	M	T	W	T	F	S
							1	2	3	4	5	6	7
9	10	11	12	13	14	15	16	17	18	19	20	21	22
23	24	25	26	27	28	29	30						

May						
S	M	T	W	F	S	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

05 Wednesday



April 2017

April 2017



Thursday 06

Week 14
April 2017

Initialization of Array

int marks[3] = {97, 95, 100};

not necessary when initialize

```
10.00
11.00
12.00
13.00
14.00
15.00
16.00
17.00
18.00
19.00
20.00
21.00
22.00
23.00
24.00
25.00
26.00
27.00
28.00
29.00
30.00
31.00
32.00
33.00
34.00
35.00
36.00
37.00
38.00
39.00
40.00
41.00
42.00
43.00
44.00
45.00
46.00
47.00
48.00
49.00
50.00
51.00
52.00
53.00
54.00
55.00
56.00
57.00
58.00
59.00
60.00
61.00
62.00
63.00
64.00
65.00
66.00
67.00
68.00
69.00
70.00
71.00
72.00
73.00
74.00
75.00
76.00
77.00
78.00
79.00
80.00
81.00
82.00
83.00
84.00
85.00
86.00
87.00
88.00
89.00
90.00
91.00
92.00
93.00
94.00
95.00
96.00
97.00
98.00
99.00
100.00
101.00
102.00
103.00
104.00
105.00
106.00
107.00
108.00
109.00
110.00
111.00
112.00
113.00
114.00
115.00
116.00
117.00
118.00
119.00
120.00
121.00
122.00
123.00
124.00
125.00
126.00
127.00
128.00
129.00
130.00
131.00
132.00
133.00
134.00
135.00
136.00
137.00
138.00
139.00
140.00
141.00
142.00
143.00
144.00
145.00
146.00
147.00
148.00
149.00
150.00
151.00
152.00
153.00
154.00
155.00
156.00
157.00
158.00
159.00
160.00
161.00
162.00
163.00
164.00
165.00
166.00
167.00
168.00
169.00
170.00
171.00
172.00
173.00
174.00
175.00
176.00
177.00
178.00
179.00
180.00
181.00
182.00
183.00
184.00
185.00
186.00
187.00
188.00
189.00
190.00
191.00
192.00
193.00
194.00
195.00
196.00
197.00
198.00
199.00
200.00
201.00
202.00
203.00
204.00
205.00
206.00
207.00
208.00
209.00
210.00
211.00
212.00
213.00
214.00
215.00
216.00
217.00
218.00
219.00
220.00
221.00
222.00
223.00
224.00
225.00
226.00
227.00
228.00
229.00
230.00
231.00
232.00
233.00
234.00
235.00
236.00
237.00
238.00
239.00
240.00
241.00
242.00
243.00
244.00
245.00
246.00
247.00
248.00
249.00
250.00
251.00
252.00
253.00
254.00
255.00
256.00
257.00
258.00
259.00
260.00
261.00
262.00
263.00
264.00
265.00
266.00
267.00
268.00
269.00
270.00
271.00
272.00
273.00
274.00
275.00
276.00
277.00
278.00
279.00
280.00
281.00
282.00
283.00
284.00
285.00
286.00
287.00
288.00
289.00
290.00
291.00
292.00
293.00
294.00
295.00
296.00
297.00
298.00
299.00
300.00
301.00
302.00
303.00
304.00
305.00
306.00
307.00
308.00
309.00
310.00
311.00
312.00
313.00
314.00
315.00
316.00
317.00
318.00
319.00
320.00
321.00
322.00
323.00
324.00
325.00
326.00
327.00
328.00
329.00
330.00
331.00
332.00
333.00
334.00
335.00
336.00
337.00
338.00
339.00
340.00
341.00
342.00
343.00
344.00
345.00
346.00
347.00
348.00
349.00
350.00
351.00
352.00
353.00
354.00
355.00
356.00
357.00
358.00
359.00
360.00
361.00
362.00
363.00
364.00
365.00
366.00
367.00
368.00
369.00
370.00
371.00
372.00
373.00
374.00
375.00
376.00
377.00
378.00
379.00
380.00
381.00
382.00
383.00
384.00
385.00
386.00
387.00
388.00
389.00
390.00
391.00
392.00
393.00
394.00
395.00
396.00
397.00
398.00
399.00
400.00
401.00
402.00
403.00
404.00
405.00
406.00
407.00
408.00
409.00
410.00
411.00
412.00
413.00
414.00
415.00
416.00
417.00
418.00
419.00
420.00
421.00
422.00
423.00
424.00
425.00
426.00
427.00
428.00
429.00
430.00
431.00
432.00
433.00
434.00
435.00
436.00
437.00
438.00
439.00
440.00
441.00
442.00
443.00
444.00
445.00
446.00
447.00
448.00
449.00
450.00
451.00
452.00
453.00
454.00
455.00
456.00
457.00
458.00
459.00
460.00
461.00
462.00
463.00
464.00
465.00
466.00
467.00
468.00
469.00
470.00
471.00
472.00
473.00
474.00
475.00
476.00
477.00
478.00
479.00
480.00
481.00
482.00
483.00
484.00
485.00
486.00
487.00
488.00
489.00
490.00
491.00
492.00
493.00
494.00
495.00
496.00
497.00
498.00
499.00
500.00
501.00
502.00
503.00
504.00
505.00
506.00
507.00
508.00
509.00
510.00
511.00
512.00
513.00
514.00
515.00
516.00
517.00
518.00
519.00
520.00
521.00
522.00
523.00
524.00
525.00
526.00
527.00
528.00
529.00
530.00
531.00
532.00
533.00
534.00
535.00
536.00
537.00
538.00
539.00
540.00
541.00
542.00
543.00
544.00
545.00
546.00
547.00
548.00
549.00
550.00
551.00
552.00
553.00
554.00
555.00
556.00
557.00
558.00
559.00
559.00
560.00
561.00
562.00
563.00
564.00
565.00
566.00
567.00
568.00
569.00
569.00
570.00
571.00
572.00
573.00
574.00
575.00
576.00
577.00
578.00
579.00
579.00
580.00
581.00
582.00
583.00
584.00
585.00
586.00
587.00
588.00
589.00
589.00
590.00
591.00
592.00
593.00
594.00
595.00
596.00
597.00
598.00
599.00
599.00
600.00
601.00
602.00
603.00
604.00
605.00
606.00
607.00
608.00
609.00
609.00
610.00
611.00
612.00
613.00
614.00
615.00
616.00
617.00
618.00
619.00
619.00
620.00
621.00
622.00
623.00
624.00
625.00
626.00
627.00
628.00
629.00
629.00
630.00
631.00
632.00
633.00
634.00
635.00
636.00
637.00
638.00
639.00
639.00
640.00
641.00
642.00
643.00
644.00
645.00
646.00
647.00
648.00
649.00
649.00
650.00
651.00
652.00
653.00
654.00
655.00
656.00
657.00
658.00
659.00
659.00
660.00
661.00
662.00
663.00
664.00
665.00
666.00
667.00
668.00
669.00
669.00
670.00
671.00
672.00
673.00
674.00
675.00
676.00
677.00
678.00
679.00
679.00
680.00
681.00
682.00
683.00
684.00
685.00
686.00
687.00
688.00
689.00
689.00
690.00
691.00
692.00
693.00
694.00
695.00
696.00
697.00
698.00
699.00
699.00
700.00
701.00
702.00
703.00
704.00
705.00
706.00
707.00
708.00
709.00
709.00
710.00
711.00
712.00
713.00
714.00
715.00
716.00
717.00
718.00
719.00
719.00
720.00
721.00
722.00
723.00
724.00
725.00
726.00
727.00
728.00
729.00
729.00
730.00
731.00
732.00
733.00
734.00
735.00
736.00
737.00
738.00
739.00
739.00
740.00
741.00
742.00
743.00
744.00
745.00
746.00
747.00
748.00
749.00
749.00
750.00
751.00
752.00
753.00
754.00
755.00
756.00
757.00
758.00
759.00
759.00
760.00
761.00
762.00
763.00
764.00
765.00
766.00
767.00
768.00
769.00
769.00
770.00
771.00
772.00
773.00
774.00
775.00
776.00
777.00
778.00
779.00
779.00
780.00
781.00
782.00
783.00
784.00
785.00
786.00
787.00
788.00
789.00
789.00
790.00
791.00
792.00
793.00
794.00
795.00
796.00
797.00
798.00
799.00
799.00
800.00
801.00
802.00
803.00
804.00
805.00
806.00
807.00
808.00
809.00
809.00
810.00
811.00
812.00
813.00
814.00
815.00
816.00
817.00
818.00
819.00
819.00
820.00
821.00
822.00
823.00
824.00
825.00
826.00
827.00
828.00
829.00
829.00
830.00
831.00
832.00
833.00
834.00
835.00
836.00
837.00
838.00
839.00
839.00
840.00
841.00
842.00
843.00
844.00
845.00
846.00
847.00
848.00
849.00
849.00
850.00
851.00
852.00
853.00
854.00
855.00
856.00
857.00
858.00
859.00
859.00
860.00
861.00
862.00
863.00
864.00
865.00
866.00
867.00
868.00
869.00
869.00
870.00
871.00
872.00
873.00
874.00
875.00
876.00
877.00
878.00
879.00
879.00
880.00
881.00
882.00
883.00
884.00
885.00
886.00
887.00
888.00
889.00
889.00
890.00
891.00
892.00
893.00
894.00
895.00
896.00
897.00
898.00
899.00
899.00
900.00
901.00
902.00
903.00
904.00
905.00
906.00
907.00
908.00
909.00
909.00
910.00
911.00
912.00
913.00
914.00
915.00
915.00
916.00
917.00
917.00
918.00
919.00
919.00
920.00
921.00
922.00
923.00
924.00
925.00
926.00
927.00
928.00
929.00
929.00
930.00
931.00
932.00
933.00
934.00
935.00
936.00
937.00
938.00
939.00
939.00
940.00
941.00
942.00
943.00
944.00
945.00
946.00
947.00
948.00
949.00
949.00
950.00
951.00
952.00
953.00
954.00
955.00
956.00
957.00
958.00
959.00
959.00
960.00
961.00
962.00
963.00
964.00
965.00
966.00
967.00
968.00
969.00
969.00
970.00
971.00
972.00
973.00
974.00
975.00
976.00
977.00
978.00
979.00
979.00
980.00
981.00
982.00
983.00
984.00
985.00
986.00
987.00
988.00
989.00
989.00
990.00
991.00
992.00
993.00
994.00
995.00
996.00
997.00
998.00
999.00
999.00
1000.00
1001.00
1002.00
1003.00
1004.00
1005.00
1006.00
1007.00
1008.00
1009.00
1009.00
1010.00
1011.00
1012.00
1013.00
1014.00
1015.00
1015.00
1016.00
1017.00
1017.00
1018.00
1019.00
1019.00
1020.00
1021.00
1022.00
1023.00
1024.00
1025.00
1026.00
1027.00
1028.00
1029.00
1029.00
1030.00
1031.00
1032.00
1033.00
1034.00
1035.00
1036.00
1037.00
1038.00
1039.00
1039.00
1040.00
1041.00
1042.00
1043.00
1044.00
1045.00
1046.00
1047.00
1048.00
1048.00
1049.00
1050.00
1051.00
1052.00
1053.00
1054.00
1055.00
1056.00
1057.00
1058.00
1059.00
1059.00
1060.00
1061.00
1062.00
1063.00
1064.00
1065.00
1066.00
1067.00
1068.00
1069.00
1069.00
1070.00
1071.00
1072.00
1073.00
1074.00
1075.00
1076.00
1077.00
1078.00
1079.00
1079.00
1080.00
1081.00
1082.00
1083.00
1084.00
1085.00
1086.00
1087.00
1088.00
1089.00
1089.00
1090.00
1091.00
1092.00
1093.00
1094.00
1095.00
1095.00
1096.00
1097.00
1097.00
1098.00
1099.00
1099.00
1100.00
1101.00
1102.00
1103.00
1104.00
1105.00
1106.00
1107.00
1108.00
1109.00
1109.00
1110.00
1111.00
1112.00
1113.00
1114.00
1115.00
1115.00
1116.00
1117.00
1117.00
1118.00
1119.00
1119.00
1120.00
1121.00
1122.00
1123.00
1124.00
1125.00
1125.00
1126.00
1127.00
1127.00
1128.00
1129.00
1129.00
1130.00
1131.00
1132.00
1133.00
1134.00
1135.00
1135.00
1136.00
1137.00
1137.00
1138.00
1139.00
1139.00
1140.00
1141.00
1142.00
1143.00
1144.00
1145.00
1145.00
1146.00
1147.00
1147.00
1148.00
1149.00
1149.00
1150.00
1151.00
1152.00
1153.00
1154.00
1155.00
1155.00
1156.00
1157.00
1157.00
1158.00
1159.00
1159.00
1160.00
1161.00
1162.00
1163.00
1164.00
1165.00
1165.00
1166.00
1167.00
1167.00
1168.00
1169.00
1169.00
1170.00
1171.00
1172.00
1173.00
1174.00
1175.00
1175.00
1176.00
1177.00
1177.00
1178.00
1179.00
1179.00
1180.00
1181.00
1182.00
1183.00
1184.00
1185.00
1185.00
1186.00
1187.00
1187.00
1188.00
1189.00
1189.00
1190.00
1191.00
1192.00
1193.00
1194.00
1195.00
1195.00
1196.00
1197.00
1197.00
1198.00
1199.00
1199.00
1200.00
1201.00
1202.00
1203.00
1204.00
1205.00
1205.00
1206.00
1207.00
1207.00
1208.00
1209.00
1209.00
1210.00
1211.00
1212.00
1213.00
1214.00
1215.00
1215.00
1216.00
1217.00
1217.00
1218.00
1219.00
1219.00
1220.00
1221.00
1222.00
1223.00
1224.00
1225.00
1225.00
1226.00
1227.00
1227.00
1228.00
1229.00
1229.00
1230.00
1231.00
1232.00
1233.00
1234.00
1235.00
1235.00
1236.00
1237.00
1237.00
1238.00
1239.00
1239.00
1240.00
1241.00
1242.00
1243.00
1244.00
1245.00
1245.00
1246.00
1247.00
1247.00
1248.00
1249.00
1249.00
1250.00
1251.00
1252.00
1253.00
1254.00
1255.00
1255.00
1256.00
1257.00
1257.00
1258.00
1259.00
1259.00
1260.00
1261.00
1262.00
1263.00
1264.00
1265.00
1265.00
1266.00
1267.00
1267.00
1268.00
1269.00
1269.00
1270.00
1271.00
1272.00
1273.00
1274.00
1275.00
1275.00
1276.00
1277.00
1277.00
1278.00
1279.00
1279.00
1280.00
1281.00
1282.00
1283.00
1284.00
1285.00
1285.00
1286.00
1287.00
1287.00
1288.00
1289.00
1289.00
1290.00
1291.00
1292.00
1293.00
1294.00
1295.00
1295.00
1296.00
1297.00
1297.00
1298.00
1299.00
1299.00
1300.00
1301.00
1302.00
1303.00
1304.00
1305.00
1305.00
1306.00
1307.00
1307.00
1308.00
1309.00
1309.00
1310.00
1311.00
1312.00
1313.00
1314.00
1315.00
1315.00
1316.00
1317.00
1317.00
1318.00
1319.00
1319.00
1320.00
1321.00
1322.00
1323.00
1324.00
1325.00
1325.00
1326.00
1327.00
1327.00
1328.00
1329.00
1329.00
1330.00
1331.00
1332.00
1333.00
1334.00
1335.00
1335.00
1336.00
1337.00
1337.00
1338.00
1339.00
1339.00
1340.00
1341.00
1342.00
1343.00
1344.00
1345.00
1345.00
1346.00
1347.00
1347.00
1348.00
1349.00
1349.00
1350.00
1351.00
1352.00
1353.00
1354.00
1355.00
1355.00
1356.00
1357.00
1357.00
1
```


Practice Qs 45 write a program to store the first n fibonacci numbers.

#include <stdio.h>

```
int main () {
```

```
    int n;
```

```
    printf("Enter n:");
```

```
    scanf("%d", &n);
```

```
    int fib[10];
```

```
    fib[0] = 0; fib[1] = 1;
```

```
    for (int i=2; i<n; i++) {
```

```
        fib[i] = fib[i-1] + fib[i-2];
```

```
    }
```

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

```
    for (int i=0; i<n; i++) {
```

```
        printf("%d\t", fib[i]);
```

```
    }
```

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

```
}
```

int max (int arr[], int n) {

```
    int max = 0;
```

```
    for (int i=0; i<n; i++) {
```

```
        if (max < arr[i]) {
```

```
            max = arr[i];
```

```
    }
```

2017	April
S M T W T F S	S M T W T F S
1 2 3 4 5 6 7 8	1 2 3 4 5 6 7 8
9 10 11 12 13 14	9 10 11 12 13 14 15
15 16 17 18 19 20	15 16 17 18 19 20 21
22 23 24 25 26 27	21 22 23 24 25 26 27
28 29 30 31	28 29 30 31

Practise Qs 46

create a 2D array, storing the tables of 2 & 3.

Done by me ✓

File No - 73

Done by me ✓

File No - 74

Done by me ✓

File No - 75

Done by me ✓

File No - 76

Done by me ✓

File No - 77

Done by me ✓

File No - 78

Done by me ✓

File No - 79

Done by me ✓

File No - 80

Done by me ✓

File No - 81

Done by me ✓

File No - 82

Done by me ✓

File No - 83

Done by me ✓

File No - 84

Done by me ✓

File No - 85

Done by me ✓

File No - 86

Done by me ✓

File No - 87

Done by me ✓

File No - 88

Done by me ✓

File No - 89

Done by me ✓

File No - 90

Done by me ✓

File No - 91

Done by me ✓

File No - 92

Done by me ✓

File No - 93

Tuesday 18

V.GUARD

April 2017



Thursday 20

19 Wednesday



Week 16
Write a program to insert an element at the end of

an array.

#include <stdio.h>

void main() {

int arr[100], n, newElement;

printf("Enter no. of elements in the array : ");

scanf("%d", &n);

printf("Enter %d elements \n", n);

for (i=0; i<n; i++) {

scanf("%d", &arr[i]);

newElement = arr[n-1];

printf("Enter new element : ");

scanf("%d", &newElement);

arr[n] = newElement;

printf("Updated array is : ");

for (i=0; i<n; i++) {

printf("%d ", arr[i]);

printf("\n");

printf("New element is : ");

printf("%d", newElement);

printf("\n");

getchar();

getchar();

getchar();

getchar();

getchar();

getchar();

getchar();

getchar();

getchar();

109-256
April 2017

CHAPTER - 8

Week 16
"Strings"

A character array terminated by a '\0' (null character).

null-character denoted string termination.

09:00 char name[] = { 'c', 'H', 'E', 'T', 'A', 'N', '\0' };

09:00 char name[] = { 'c', 'H', 'E', 'T', 'A', 'N', '\0' } //GUESS

09:00 char village[] = { 'q', 'U', 'D', 'A', 'S', '\0' } //QUTADS

If we forgot to write '\0' within It is not a string.
it acts as char array. but only first 6 letters are printed.

diff between strings and character array Is strings have special properties to apply, operations

09:00 Initialize strings.

char name[] = { 'c', 'H', 'E', 'T', 'A', 'N', '\0' };

char name[] = "CHETAN"; it will complies automatically '\0' is added

April						
S	M	T	W	F	S	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

May						
S	M	T	W	F	S	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
21	22	23	24	25	26	27
28	29	30	31			

April 2017

April 2017

Saturday 22



V-GUARD
Week 16

21 Friday

char village[] = "GUJARAT";

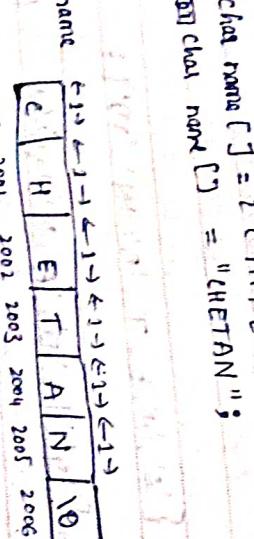
When declare a string in memory?

What happens in memory?

char name[] = {'C','H','E','T','A','N','\0'};

char name[] = "CHETAN";

char name[] = {C,H,E,T,A,N,\0};



Practice Ques 47

File No - 16

Create a string firstName and lastName to store details of

user and print all the characters using a loop

char firstName[] = "CHETAN";

char lastName[] = "INHALI";

#include <stdio.h>
void printString (char arr[]); // size not required because know
int main(){
 strings ends with '\0'
 char firstName[] = "CHETAN";
 char lastName[] = "INHALI";
 printString (firstName);
 printString (lastName);
}

2017 April May
S M T W T F S S M T W T S
1 2 3 4 5 6 7 8 9 10 11 12 13
14 15 16 17 18 19 20 21 22 23 24 25 26 27
28 29 30 31

Sunday 23



Week 16

void printString (char arr[]){
 for (int i=0 ; arr[i] != '\0' ; i++) {
 printf ("%c\n", arr[i]);
 }
}

printf ("\n");

File No - 16

for (int i=0 ; arr[i] != '\0' ; i++) {
 printf ("%c\n", arr[i]);
}

printf ("\n");

April 2017

V-GUARD

24 Monday

String Format specifier

Week 17

```
00:00
"%.5s"
01:00
char name[] = "CHETAN";
02:00
printf ("%s", name);
```

```
03:00
char name[40];
04:00
scanf ("%s", name); // fname
```

Because Array kudh Ek pointer hai

05:00

Practice Qs 18
Ask the user to enter their firstName & print it back to them.
Also try this with their full name (fullname) - Multi word string

Imp:
scanf cannot input multiple-word strings with spaces
gets() & puts() come into picture

2017	April
S M T W T F S	S M T W T F S
1 2 3 4 5 6 7	8 9 10 11 12 13
9 10 11 12 13 14	15 16 17 18 19 20
21 22 23 24 25 26 27	28 29 30 31

Tuesday 25

string Functions

Week 17

```
00:00
gets(str) → Dangerous & Outdated
01:00
input a string even a multiword
02:00
maxsize(x). Enter input
03:00
```

04:00
fgets(Cstr,n,file) file "stdin"

05:00
takes n-1 characters from file or if n more than n-1 char entered

06:00
gets(str) → Dangerous & Outdated
07:00
gets(str) → Dangerous & Outdated
08:00
gets(str) → Dangerous & Outdated
09:00
gets(str) → Dangerous & Outdated
10:00
gets(str) → Dangerous & Outdated

V-GUARD

2017	May
S M T W T F S	S M T W T F S
1 2 3 4 5 6 7	8 9 10 11 12 13
14 15 16 17 18 19	20 21 22 23 24 25 26
27 28 29 30 31	

26 Wednesday



April 2017

String using Pointers

char array
 \uparrow
 string \rightarrow array \rightarrow pointer

char *str = "Hello world";
 \uparrow
 str → preassigned address is stored in
 the char pointer 'str'

store string in memory preassigned address is stored in
 the char pointer 'str'

④ char *str = "Hello World"; // can be reinitialized
 \downarrow
 char *str = "Hello";
 \downarrow
 modified

→ but copying possible in strcpy
 but cannot be reinitialized
 modified

Practice Qs 49
 make a program that inputs user's name & prints its length.
 Done by me ↴

int countLength (char arr[]);

int main () {

char name [100];

gets (name, 100, stdin); // "CNAME\n\0."

printf ("Length is: %d", countLength (name));
 }

int countLength (char arr[]) {

int count = 0;

for (int i = 0; arr[i] != '\0'; i++) {
 \uparrow
 count++;

return count - 1; \rightarrow because fgets adds \n at last

April 2017



Thursday 27

strings problems: maximum solved by loops \rightarrow concat arr[1] = '0'

Standard Library Functions

#include <iostream.h>
 #include <string.h>

1. strlen (str) → string length.
 count no. of characters excluding '\0'

#include <iostream.h>
 #include <string.h>

int main () {
 char name [] = "CHETAN";

cout << "Length is : " << strlen (name);
 cout << endl;

printf ("Length is : %d", length); // 6

2. strcpy (newStr, oldStr) → string copy
 copies the value of old string to new string

int main () {

char oldStr [] = "oldstr";

char newStr [] = "newstr";

strcpy (newStr, oldStr);
 puts (newStr); \downarrow
 // oldstr

Possible as long as newStr.

has enough space
 (i.e. length newStr $>$ oldStr)

April						
S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

May						
S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

01 Monday



May 2017

May 2017



Tuesday 02

Week 18

Practice Qs 52 : Write a function named slice, which takes a string & returns a sliced string from index n to m.

09:00 slice("HelloWorld", 3, 6) → Hello

10:00 slice("HelloWorld", 3, 6) → file.No - 83

11:00 slice("HelloWorld", 3, 6) → file.No - 85

12:00 #include <stdio.h>

void main() { char str[], int n, int m); }

char str[] = "Hello World";

void slice(char str[], int n, int m) {

char newStr[100];

int j = 0; for (int i = 0; i < m; i++) { newStr[j] = str[i]; j++; }

newStr[j] = '\0'; puts(newStr); }

{ #include <stdio.h> void main() { char str[] = "HelloWorld"; puts(str); } }

(d) write a program to replace lowercase letters with uppercase letters

of a string

Solver: charConvolution.c

10th file

2017 May

2017 June

S M T W T F S S M T W T F S

1 2 3 4 5 6 7 8 9 10 11 12 13

11 12 13 14 15 16 17 18 19 20 21 22 23

24 25 26 27 28 29 30

28 29 30 31



May 2017

May 2017



Practice Qs 53 : Q) occurrence of vowels in a strings. (Done by me) File No - 84

09:00 void str[i] == 'a' || str[i] == 'e' || str[i] == 'i' || str[i] == 'o' || str[i] == 'u'

10:00 Practice Qs 54 :

11:00 & check if a given character is present in a string or not.

12:00 void: File.No - 85 (Done by me)

* String length using pointers

01:00 Homework Set

02:00 Q) write a program to convert all lowercase letters vowels to uppercase in a string

#include <stdio.h> void charConvolution.c // 8th file

03:00 Q) write a program to print the highest frequency char in a string

04:00 Soln: highFrequencyCharInString // 8th file

using freq[]

05:00 Q) write a program to remove blank spaces in a string

06:00 Soln: removeBlankSpaces.c // 8th file

(c) write a program to replace lowercase letters with uppercase letters

Solver: charConvolution.c

10th file

2017 May

2017 June

S M T W T F S S M T W T F S

1 2 3 4 5 6 7 8 9 10 11 12 13

11 12 13 14 15 16 17 18 19 20 21 22 23

24 25 26 27 28 29 30

28 29 30 31