11/09/24

C. Programming

Theory Assignment 1

Submitted by: Sakshom Srivastava
Rank: 353, BCA 1-C

DELTA | Pg No | Date / /

••• Questions:

Ques-1:) Write a program to show all the prime numbers within a user-specified range.

Ans-1:)

```
1.  #include<stdio.h>
2.  void main(){
3.      printf("Program to find Prime Numbers \n");
4.      int n,i,j,p;
5.      printf("Enter No. upto which you want to find:");
6.      scanf("%d",&n);
7.      for(i=2;i<=n;i++){
8.          for(j=2;j<i;j++){
9.              if(i%j!=0){
10.                 p=i;}
11.             else{
12.                 p=0;
13.                 break;}
14.     }
15.     if(p!=0){
16.         printf("\n Prime Number is %d",p);
17.     }
18. }
19. }
```

→ Line ↓

Dry Run:

line:3. Program to find Prime Numbers

line:6. n = 6

line:7 for1 : i=2, 2<6

line:8 ↳ for2 : j=2, 2<2 // false, back to line 7, i++=3

line:7 for1 : i=3, 3<6

line:8 ↳ for2 : j=2, 2<3

line:9 ↳ (3%2 != 0) // true, So p=i=3 | back to line 8, j++=3

line:8 for2 : j=3, 3<3 // false, out of loop2 @ line 15

line:15 (p!=0) i.e. (3!=0) // true So print "3 is prime No."

line 7 for1 : i=4, 4<6

line 8 ↳ for2 : j=2, 2<4

line 9 ↳ (4%2 != 0) // false, execute else block

line 12 p=0, break for loop 2, @ line 15 / line 8: j=5, 5<5 // false (break)

line 15 (p!=0) // false, back to line 7

line 7 for1 : i=5; 5<6

line 8 ↳ for2 : j=2, 2<5

line 9 ↳ (5%2 !=0) // true p=i=5

line8 ↳ for2 : j=3, 3<5

line9 ↳ (5%3 !=0) // true p=i=5

line8 ↳ for2 : j=4, 4<5

line9 ↳ (5%4 !=0) // true p=i=5

Basic Logic:

- check remainders when 'n' divided by every number less than 'n' >1;
- if none remainder is '0', it is a Prime Number.

line 15 : p!=0 as p=5
∴ printf 5 is prime

i=6, j=2
(6%2 != 0) // false
break

// end of program.

DELTA | Pg No | Date / /

Ques-2:) Write a program to check whether a number is an Armstrong Number or Not.

Armstrong No: $\underbrace{153}_{3} = 1^3 + 5^3 + 3^3 = 153$

Ans-2:)

```
1.  #include<stdio.h>
2.  # define YES  printf("%d is an Armstrong No.")
3.  # define NO   printf("%d is NOT an Armstrong No.")
4.  ;
5.  void main(){
6.      printf("Program to find Armstrong Number");
7.      int n, og, i, d, a=0;
8.      printf("\nEnter Number:");
9.      scanf("%d", n);
10.     og = n
11.
12.     for(i=1; i<=3; i++){
13.         d = n%10;
14.         n /= 10;
15.         a += d*d*d
        }
16.     }
17.     (a==og) ? YES : NO;
18. }
```

→ Dry Run:

line:6 Program to find armstrong No.
line:9 n = 153
line:10 og = n = 153
line:12 for 1: i=1, 1<3
line:13 ↳ (d = 153%10) d=3
line:14 (n = 153/10) n=15
line:15 (a = a + 3x3x3) = 0+27=27 // i++ @ line 12

line:12 for 1: i=2, 2<3
line:13 ↳ (d = 15%10) d=5
line:14 (n = 15/10) n=1
line:15 (a = a + 5+5x5) = 27+125=152 // i++ @ line 12

line:12 for 1: i=3, 3==3
line:13 ↳ (d = 1%10) d=1
line:14 (n = 1/10) n=0
line:15 (a = a+1x1x3) = 152+1=153 // i++ @ line 12

line:12 for 1: i=4, 4>3 // break @ line 17

line:17 (a == og) i.e. (153 == 153) // true

★ in preprocessing 'YES' is a macro. so, print "153 is on Armstrong No."

Basic Logic:
- break 'n' into individual digits
  ↳ add the cube of digits.
- If the result is equal to orignal 'n' then it is on armstrong Number.

DELTA Page No.
Date

Ques-3:) Write a program to find Fibonacci Series of ~~first~~ 'n' elements.

Fibonacci Numbers = 0+1 = 2 ! 3+5 = 8 ! 13+21 => 34
1+2 = 3 ! 5+8 = 13 ! 21+34 => 55
2+3 = 5 ! 8+13 = 21 ! 34+55 => 89

Ans-3:)

```
1.  #include<stdio.h>
2.  void main(){
3.      printf("Program to print Fibonacci Series \n");
4.      int n, i, t, a=0, b=1;
5.      printf("Enter Number of Fibonacci Elements : ");
6.      scanf("%d", &n);
7.      printf("\n Fibonacci Series : ");
8.      printf("\n%d \n%d", a, b);
9.
10.     for(i=1; i<=(n-2); i++){
11.         f = a+b;
12.         a = b;
13.         b = f;
14.         printf("\n%d", f);
15.     }
16. }
```

→ Dry Run:

line 3 Program to find Fibonacci Series

line 6 n = 5

line 8, 0 ← this will get printed
1

line 10 for i : i = 1, 1 < (5-2)
line 11 ↳ (f = a+b) = 0+1 = f = 1
line 12 a = b = 1
line 13 b = f = 1
line 14 print f i.e 1, loop i++

line 10 for i: i = 2, 2 < (5-2)
line 11 ↳ (f = a+b) = 1+1 = f = 2
line 12 (a = b = 1)
line 13 (b = f = 2)
line 14 print f i.e 2, loop i++

line 11 for i: i = 3, 3 = (5-2)
line 12 ↳ (f = a+b) = 2+2 = 3
line 13 (a = b = 2)
line 14 (b = f = 3)
line 15 print f i.e. 3, loop i++

line 10 for i : i = 4 4 > (5-3) // break, end-of-program

Basic Logic
- a = 0, b = 1
- add a+b
- transfer value of b in a and value of current fibonacci/sum in b.
- Loop Again

DELTA | Pg No
Date / /

Ques-4:) Write a program to print Fibonacci Series using Recursion in user defined function.

Ans-4:)

```
1.  # include<stdio.h>
2.  void main(){
3.  printf("Program to print Fibonacci Series \n");
4.  int n, a=0, b=1, i=1;
5.  printf("Enter Number of Fibonacci Elements : ");
6.  scanf("%d", &n);
7.  printf("\n Fibonacci Series : ");
8.  printf("\n%d\n%d", a, b);
9.  fibonacci(a,b,n,i);
    }
10. }
11. void fibonacci(int a, int b, int n, int i){
12.     int f;
13.     f = a+b;
14.     a = b;
15.     b = f;
16.     if(i==(n-1)){
17.     return ; }
18.     else{
19.     printf("\n%d", f);
20.     i++;
21.     fibonacci(a, b, n, i);
22.     }
23. }
```

→ Dry Run :

line 6 | n = 5

line 9 | in fiboracci function passing orguements n=5, a=0, b=1, i=1

line 13 | (f = a+b) = 0+1, f = 1

line 14 | (a=b=1)

line 15 | (b=f=1)

line 16 | if (i == (5-1)) i.e (1 == 4) // false else statement

line 19 | printed f i.e 1 // i++ = 2, Recorrsion call() ← new values in = a, b, i

line 13 | (f = a+b) = 1+1, f = 2

line 14 | (a=b=1)

line 15 | (b=f=2)

line 16 | if (i == (5-1)) i.e. (2 == 4) // false else statement

line 19 | printed f i.e 2 // i++ = 3, Recorrsion call() again new values in = a, b, i

line 13 | (f = a+b) = 1+2 ; f = 3

line 14 | (a=b=2)

line 15 | (b=f=3)

line 16 | if (i == (5-1)) i.e (3 == 4) // false else statement

line 19 | printed f i.e 3 // i++ = 4, Recorrsion call() again new values in = a, b, i

line 16 | here if (i == (5-1)) ie. (4 == 4) // true ∴ return to line 9

end-of-program