C Practical

-Shrisai Halankar

Pattern 1:

|  |  |
| --- | --- |
| Algorithm | Dry Run |
| 1. Start  2. Declare integer i and j  3. Input  3.1 Use for loop and declare i=1,i<=3,i++  3.2 Use for loop and declare j=1,j<=4,j++  4. Print \*  5. Print /n for line spacing  6. Stop Program | |  |  |  | | --- | --- | --- | | Step | i | j | | 1  2  3  4  5  6  7  8  9  10  11  12 | 1  1  1  1  2  2  2  2  3  3  3  3 | 1  2  3  4  1  2  3  4  1  2  3  4 | |

Source code:

#include<stdio.h>

#include<conio.h>

void main ()

{

/\* function declaration \*/

int i,j;

for(i=1 ; i<=3;i++)

{

for(j=1;j<=4;j++)

printf("\*");

printf("\n");

}

getch();

}

Output:



Pattern 2:

|  |  |
| --- | --- |
| Algorithm | Dry Run |
| 1. Start  2. Declare integers i,j and k  3. Input  3.1. Use for loop and declare i=1,i<=4,i++  3.2. Use for loop and declare j=4-I,j>0,j—  3.3. Print Blank space by doing(“ “)  4. Use for loop and declare k=1,k<=I,k++  5. Print \*  5.1 Print /n for line spacing  6. Stop | |  |  |  |  | | --- | --- | --- | --- | | Step | i | j | k | | 1  2  3 | 1  2  3 | 3  2  1 | 1  2  3 | |

Source Code:

#include<stdio.h>

#include<conio.h>

void main()

{

char i,j,k;

for(i=1;i,i<=4;i++)

{

for(j=4-i;j>0;j--)

{

printf(" ");

}

for(k=1;k<=i;k++)

{

printf("\* ");

}

printf("\n");

}

getch();

}

Output:



Pattern 3:

|  |  |
| --- | --- |
| Algorithm | Dry Run |
| 1. Start  2. Declare integers i,j and k  3. Use for loop and declare i=1,i<5,i++  3.1 Use for loop j=0,j<=i-1,j++  3.1.2 For spacing use (“ “)  4. Use for loop k=4,k>=i,k—  4.1. Print \*  5. Print /n for line spacing  6. Stop | |  |  |  |  | | --- | --- | --- | --- | | Step | i | j | k | | 1  2  3  4  5 | 1  2  3  4  5 | 0  1  2  3  4 | 4  3  2  4  5 | |

Source Code:

#include<stdio.h>

#include<conio.h>

void main()

{

char i,j,k;

for(i=1;i<5;i++)

{

for(j=0;j<=i-1;j++)

{

printf(" ");

}

for(k=4;k>=i;k--)

{

printf("\* ");

}

printf("\n");

}

getch();

}

Output:



Pattern 4:

|  |  |
| --- | --- |
| Algorithm | Dry Run |
| 1. Start  2. Declare i,j  3. for(i=1;j<=i;j++)  Print→\n  a. print(j=1;j<=i;j++)  print→\n  4. end |  |

Source Code:

#include<stdio.h>

#include<conio.h>

void main()

{

char i,j;

for(i=1;i<5;i++)

{

for(j=1;j<=i;j++)

{

printf("\* ");

}

printf("\n");

}

getch();

}

Output:



5. Fibonacci Series

|  |  |
| --- | --- |
| Algorithm | Dry Run |
| 1. Start  2. Declare no,sum=1,n1=0,n+1  3. While(sum<=no)  a. Print sum  b. sum=n1+n2  c. n1=n2  d. n2=sum  4. End |  |

Source Code:

#include <stdio.h>

int main()

{

int a, b, c, i, n;

a = 0;

b = 1;

printf("Enter a number to define the length of fibonacci series: ");

scanf("%d", &n);

printf("\nThe Series is: \n");

printf("%d\t%d", a, b);

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

{

c = a + b;

a = b;

b = c;

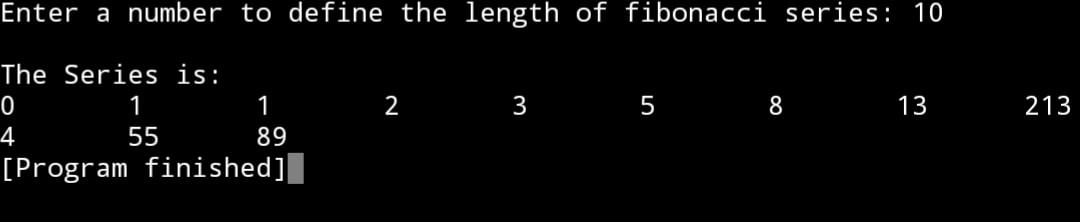
printf("\t%d", c);

}

return 0;

}

Output:



6. Sum of Factorials:

|  |  |
| --- | --- |
| Algorithm | Dry Run |
| 1. Start  2. Declare  no,j=1,i=1,product=1,sum=0  3. While(i<=no)  a. while(j<=i)  i. pd=pd\*j  ii. j=j+1  b. sum=sum+pd  4. End |  |

Source Code:

// This is a program to add the factorial of a series

#include<stdio.h>

#include<conio.h>

void main()

{

int no,j=1,i=1,product=1,sum=0;

printf("Enter the number: ");

scanf("%d",&no);

while(i<=no)//adding all factorial

{

while(j<=i)

{

product=product\*j;

j=j+1;

}

sum=sum+product;

i=i+1;

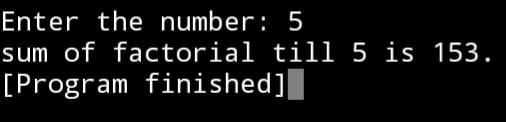
}

printf("sum of factorial till %d is %d.",no,sum);

getch();

}

Output:



7. Prime number :

|  |  |
| --- | --- |
| Algorithm | Dry Run |
| 1. Start  2. Declare  no,div,i=2,sum=0,prime  3. Input no  4. while(i<=no)  5. prime=1  6. for(div=2;div<=i/2;d++)  a. if(i5div==0)  prime=0  break;  b. if(prime==0)  sum=sum+1  c. i=i+1  7. Print sum  8. End |  |

Source code:

// This is a program to add prime number till n

#include<stdio.h>

#include<conio.h>

void main()

{

int no,div,i=2,sum=0,prime;

printf("Enter the number: ");

scanf("%d",&no);

while(i<=no)

{

prime=1;

for(div=2;div<=i/2;div++)

{

if(i%div==0)

{

prime=0;

break;

}

}

if (prime==1)

{

sum=sum+i;

}

i=i+1;

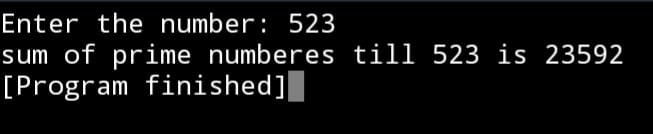
}

printf("sum of prime numberes till %d is %d",no,sum);

getch();

}

Output:



8. nCr nPr:

|  |  |
| --- | --- |
| Algorithm | Dry Run |
| 1. Start  2. Declare  Int no,i,j,r  Float nCr,nPr  Long int fact=1,fact\_r,fact\_no\_r  Source code:  #include<stdio.h>  #include<conio.h>  void main()  {  int no,i,j,r;  3. for(i=r;i<=j;i++)  a. if(i==r)  fact\_r=fact  b. aif(i==(no-r))  fact\_no\_r=fact  4. nCr=fact/fact\_no\_r  5. nPr=fact/(fact\_no\_r\*fact\_r)  6. Print nCr and nPr  7. end |  |

Source code:

// This is a program to finf nCr and nPr

#include<stdio.h>

#include<conio.h>

void main()

{

int no,r,i;

float nCr,nPr;

long int fact\_no=1,fact\_no\_r=1,fact\_r=1;

printf("Enter the value of no and r: ");

scanf("%d%d",&no,&r);

for (i=1; i<=no; i++)

{

fact\_no=fact\_no\*i;

}

for (i=1; i<=r; i++)

{

fact\_r=fact\_r\*i;

}

for (i=1; i<=(no-r); i++)

{

fact\_no\_r=fact\_no\_r\*i;

}

nCr=(float)(fact\_no/fact\_no\_r);

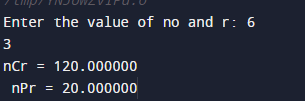
nPr=(float)(fact\_no/(fact\_no\_r\*fact\_r));

printf("nCr = %f \n nPr = %f",nCr,nPr);

getch();

}

Output:



9. Sum of Natural numbers:

|  |  |
| --- | --- |
| Algorithm | Dry Run |
| 1. Start  2. Declare no, sum=0  3. Input &no  4. While  a. sum=sum+no  b. no=no-1  5. Print sum  6. end |  |

Source code:

#include<stdio.h>

#include<conio.h>

void main()

{

int n,i,sum=0;

printf("Enter a positive number: ");

scanf("%d",&n);

i=1;

while(i<=n) {

sum+=i;

++i;

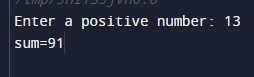
}

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

getch();

}

Output:



10. Prime Number or Not:

|  |  |
| --- | --- |
| Algorithm | Dry Run |
| 1. Start  2. Declare no,div=2  3. Input no  4. if (no<2)  a. Not prime  5. Else (while)  a. If (no%d==0)  i. Not prime  b. div=div+1  6. if (div==no)  a. Prime.  7. End |  |

Source code:

#include<stdio.h>

#include<conio.h>

void main ()

{

int no,div=2;

printf("Enter a number: ");

scanf("%d",&no);

if (no<2)

{

printf("%d is not a prime number",no);

}

else

{

while (div<no)

{

if (no%div==0)

{

printf("%d is not a prime number",no);

break;

}

div=div+1;

}

if (div==no)

{

printf("%d is a prime number",no);

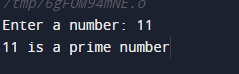
}

}

getch();

}

Output:



11. Factorial of a Number

|  |  |
| --- | --- |
| Algorithm | Dry Run |
| 1. Start  2. Declare  no,product=1,lap=0  3. Input no  4. if (no<0)  a. No factorial  5. if (no==0)  a. fact=1  6. else(while)(lap<no)  a. lap=lap+1  b. pdt=pdt\*lap  c. if(lap=no)  i. Fact  7. end |  |

Source code:

#include<stdio.h>

#include<conio.h>

void main()

{

int no, product=1,lap=0;

printf("Enter a number: ");

scanf("%d",&no);

while (lap<no)

{

lap=lap+1;

product=product\*lap;

if(lap==no)

{

printf("Factorial of %d is %d.",no,product);

break;

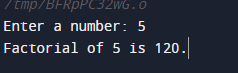
}

}

getch();

}

Output:



12. Reverse and Add

|  |  |
| --- | --- |
| Algorithm | Dry Run |
| 1. Start  2. Declare  no,sum=0,rev=0,rem  3. While  a. rem=no%10  b. rev=rev\*10+rem  c. no=no/10  d. sum=sum+rem  4. Print rev,sum  5. End |  |

Source code:

#include<stdio.h>

#include<conio.h>

void main()

{

int no, sum=0, rev=0, rem;

printf("Enter a number: ");

scanf("%d",&no);

while(no!=0)

{

rem=no%10;

rev=rev\*10+rem;

no=no/10;

sum=sum+rem;

}

printf("reverse = %d \n sum = %d",rev,sum);

getch();

}

Output:

