**PF Lab No: 06**

**Name: Ali Rooman**

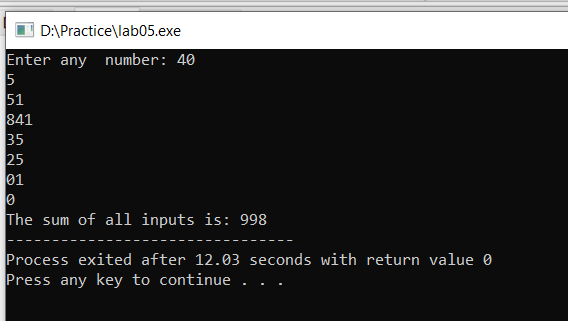
**Roll No: 24K-0792**

1. Which loop system would be better for user input. Justify your answer by creating a program that takes a value and adds it to a variable and prints it repeatedly until the user enters a zero value.

Ans. Do while loop will be better to use in this context as because it guarantees at least one iteration, which is often expected behavior when asking for user input in an interactive program.

|  |
| --- |
| #include <stdio.h>  int main(){  int sum = 0, input;  printf("Enter any number: ");  do  {  scanf("%d",&input);  sum = sum + input;  }  while(input!=0);  printf("The sum of all inputs is: %d",sum);    return 0;  } |

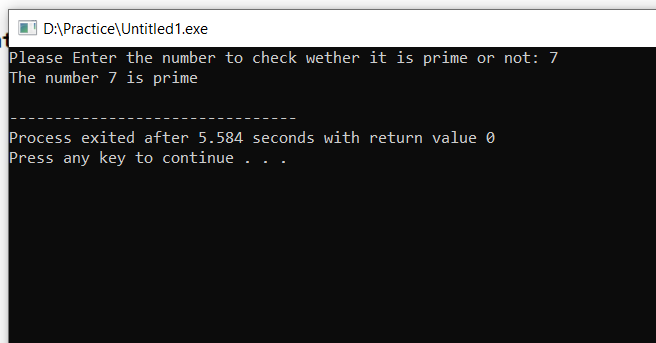
**OUTPUT**

****

2. Write a program to check whether a given number is prime or not.

|  |
| --- |
| #include <stdio.h>  #include <math.h>  #include <stdbool.h>  int main() {  int i, n;  bool isPrime = true;  printf("Please Enter the number to check wether it is prime or not: ");  scanf("%d", &n);  if (n <= 1) {  printf("The number %d is not prime\n",n);  } else {  for (i = 2; i <= sqrt(n); i++) {  if (n % i == 0) {  isPrime = false;  break;  }  }  if (isPrime) {  printf("The number %d is prime\n", n);  } else {  printf("The number %d is not prime\n", n);  }  }  return 0;  } |

**OUTPUT**

****

3. Using the above program integrate the number if it is a prime and print the Fibonacci series till that

number.

Example

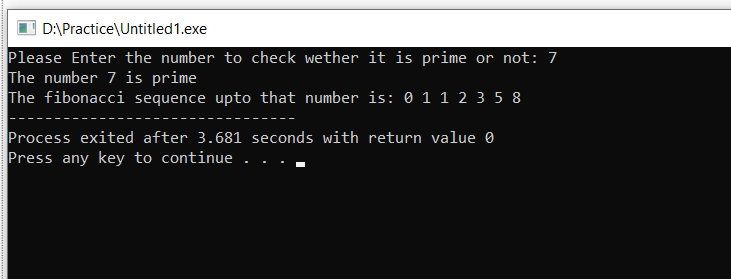
Input: 5 Output:

Number is prime

Series is = 0 1 1 2 3

|  |
| --- |
| #include <stdio.h>  #include <math.h>  #include <stdbool.h>  int main() {  int i, n;  bool isPrime = true;  printf("Please Enter the number to check whether it is prime or not: ");  scanf("%d", &n);  if (n <= 1) {  printf("The number %d is not prime\n",n);  } else {  for (i = 2; i <= sqrt(n); i++) {  if (n % i == 0) {  isPrime = false;  break;  }  }  if (isPrime) {  printf("The number %d is prime\n", n);  int t1=0, t2=1, next term;  int j;  printf("The Fibonacci sequence up to that number is: ");  for(j=1;j<=n;j++)  {  printf("%d ",t1);  nextterm = t1 + t2;  t1 = t2;  t2 = nextterm;  }  } else {  printf("The number %d is not prime\n", n);  }  }  return 0;  } |

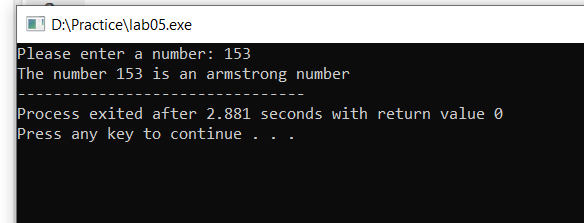
**OUTPUT**

****

4. Write a program to check whether a number is an Armstrong number or not. An Armstrong number is a number that is equal to the sum of cubes of its digits.

|  |
| --- |
| #include <stdio.h>  int main(){  int num, remainder,result = 0;  printf("Please enter a number: ");  scanf("%d", &num);  int orignal\_num = num;  while(num !=0){  remainder = num%10;  result = result + (remainder \* remainder \* remainder);  num = num/10;  }  if(orignal\_num==result){  printf("The number %d is an armstrong number",orignal\_num);  }else{  printf("The number %d is not an armstrong number",orignal\_num);  }  return 0;  } |

**OUTPUT**

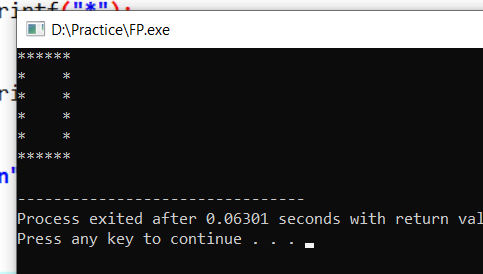
****

5.Make a pattern mentioned below using loops

Pattern 1

|  |
| --- |
| #include <stdio.h>  int main() {  int i,j;  for(i=1;i<=6;i++){  for(j=1;j<=6;j++){  if(i==1||i==6 || j==1 || j==6){  printf("\*");  }  else{  printf(" ");  }  }  printf("\n");  }  return 0;  } |

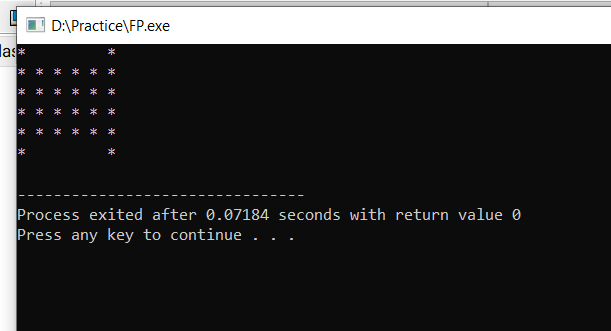
OUTPUT



Pattern 2

|  |
| --- |
| #include <stdio.h>  int main() {  int n = 6;  int i,j;  for (i = 0; i < n; i++) {  for (j = 0; j < n; j++) {  if (i == 0 || i == n - 1) {  if (j == 0 || j == n - 1) {  printf("\* ");  } else {  printf(" ");  }  } else {  if (i % 2 == 1) {  printf("\* ");  } else {  printf("\* ");  }  }  }  printf("\n");  }  return 0;  } |

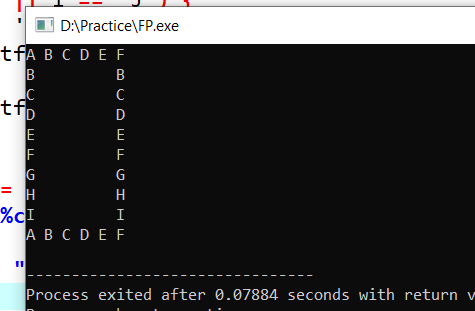
OUTPUT



Pattern 3

|  |
| --- |
| #include <stdio.h>  int main() {  int i, j;  for (i = 'A'; i <= 'J'; i++) {  for (j = 'A'; j <= 'J'; j++) {  if (i == 'A' || i == 'J') {  if (j <= 'F') {  printf("%c ", j);  } else {  printf(" ");  }  }  else if (j == 'A' || j == 'J') {  printf("%c ", i);  } else {  printf(" ");  }  }  printf("\n");  }  return 0;  } |

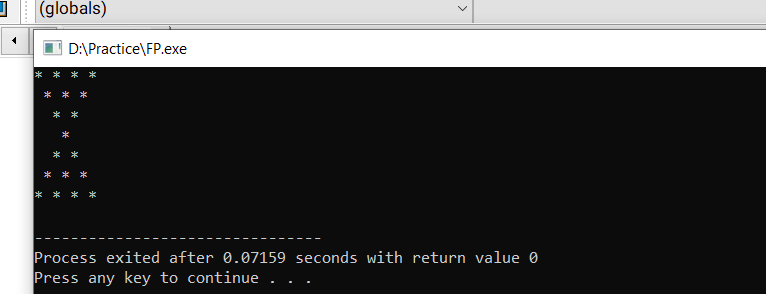
OUTPUT



Pattern 4

|  |
| --- |
| #include <stdio.h>  int main() {  int n = 4;  for (int i = 0; i < n; i++) {  for (int j = 0; j < i; j++) {  printf(" ");  }  for (int j = 0; j < n - i; j++) {  if (j == 0 || j == n - i - 1) {  printf("\* ");  } else {  printf("\* ");  }  }  printf("\n");  }  for (int i = 1; i < n; i++) {  for (int j = 0; j < n - i - 1; j++) {  printf(" ");  }  for (int j = 0; j < i + 1; j++) {  if (j == 0 || j == i) {  printf("\* ");  } else {  printf("\* ");  }  }  printf("\n");  }  return 0;  } |

OUTPUT



Pattern 5

|  |
| --- |
| #include <stdio.h>  int main() {  int n = 4;  char ch = 'A';  for (int i = 0; i < n; i++) {  for (int j = 0; j < i; j++) {  printf(" ");  }  for (int j = 0; j < n - i; j++) {  printf("%c ", ch);  ch++;  }  printf("\n");  }  for (int i = 1; i < n; i++) {  for (int j = 0; j < n - i - 1; j++) {  printf(" ");  }  for (int j = 0; j < i + 1; j++) {  printf("%c ", ch);  ch++;  }  printf("\n");  }  return 0;  } |

OUTPUT

