**Assignment N0-6**

1. Write a program to display all prime numbers between 1 and 100.

#include <stdio.h>

int main() {

int i, j, isPrime;

printf("Prime numbers between 1 and 100 are:\n");

for (i = 2; i <= 100; i++) { // start from 2, since 1 is not a prime

isPrime = 1; // assume i is prime

for (j = 2; j <= i / 2; j++) { // check divisibility

if (i % j == 0) {

isPrime = 0; // not prime

break;

}

}

if (isPrime == 1) {

printf("%d ", i);

}

}

return 0;

}

2. Write a program to display all Armstrong numbers between 1 and 500. (An Armstrong number is a number such that the sum of cube of digits = number itself Ex. 153 = 1\*1\*1 + 5\*5\*5 + 3\*3\*3

#include <stdio.h>

int main() {

int num, original, remainder, sum;

printf("Armstrong numbers between 1 and 500 are:\n");

for (num = 1; num <= 500; num++) { // check each number from 1 to 500

original = num; // store original number

sum = 0; // reset sum

while (original != 0) { // extract each digit

remainder = original % 10; // get last digit

sum += remainder \* remainder \* remainder; // add cube of digit

original /= 10; // remove last digit

}

if (sum == num) { // if sum of cubes equals original number

printf("%d ", num); // print Armstrong number

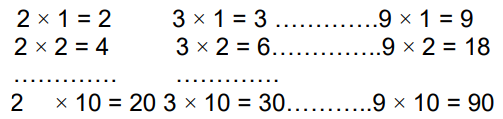
}

}

return 0; // exit program

}

3. Write a program to display multiplication tables from to having n multiples each. The output should be displayed in a tabular format. For example, the multiplication tables of 2 to 9 having 10 multiples each is shown below.



#include <stdio.h>

int main() {

int start, end, multiples, i, j;

printf("Enter starting number of table: ");

scanf("%d", &start);

printf("Enter ending number of table: ");

scanf("%d", &end);

printf("Enter number of multiples for each table: ");

scanf("%d", &multiples);

printf("\nMultiplication tables from %d to %d:\n\n", start, end);

for (i = 1; i <= multiples; i++) { // loop for each multiple (row)

for (j = start; j <= end; j++) { // loop for each table (column)

printf("%d x %d = %d\t", j, i, j \* i); // print one value with tab space

}

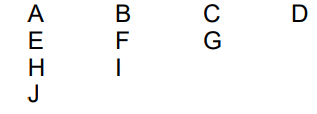
printf("\n"); // new line after each row

}

return 0;

}

4. Modify the sample program 1 to display n lines as follows (here n=4).



#include <stdio.h>

int main() {

int n, i, j;

char ch = 'A'; // start with letter A

printf("Enter number of lines (n): ");

scanf("%d", &n);

for (i = n; i >= 1; i--) { // loop for number of rows (from n down to 1)

for (j = 1; j <= i; j++) { // loop for each column in that row

printf("%c ", ch); // print current character

ch++; // go to next character

}

printf("\n"); // new line after each row

}

return 0;

}