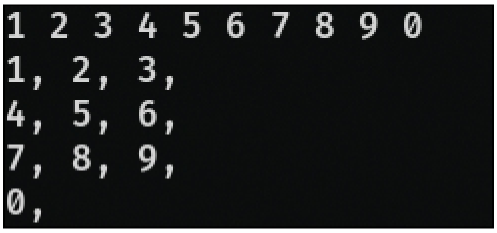
Name: Yashwant Chandrakant Bhosale
MIS: 612303039
Div: SY COMP DIV1

Control statements assignment

1. Write a program to read 10 integers. Display these numbers by printing three numbers in a line separated by commas.
Code:

```
#include <stdio.h>

int main() {
    int arr[10];
    for(int i = 0; i < 10; i++) {
        scanf("%d", &arr[i]);
    }
    for(int j = 0; j < 10; j++) {
        printf("%d", arr[j]);
        if(!((j+1)%3)) printf(",\n");
        else printf(", ");
    }
    printf("\n");
    return 0;
}
```

output:

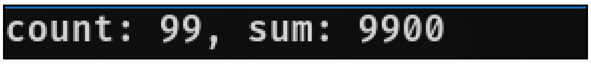


```
1 2 3 4 5 6 7 8 9 0
1, 2, 3,
4, 5, 6,
7, 8, 9,
0,
```

2. Write a program to print the count of even numbers between 1–200. Also print their sum.
Code:

```
#include <stdio.h>

int main() {
    int count = 0, sum = 0;

    for(int i = 1; i < 200; i++) {
        if(!(i%2)) {
            sum += i;
            count++;
        }
    }
    printf("count: %d, sum: %d\n", count, sum);
    return 0;
}
```

Output:

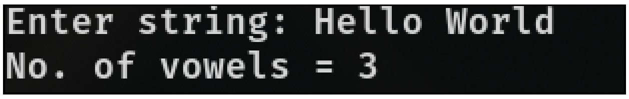


```
count: 99, sum: 9900
```

3. Write a program to count the number of vowels in a text. (eg: Enter text: hello world,
Output: No.of vowels- 3).
Code:

```
#include <stdio.h>

int count_vowels(char *str) {
    int i = 0, count = 0;
    while(str[i] != '\0') {
        if(str[i] == 'a' || str[i] == 'A' || str[i] == 'e' || str[i] ==
'E' || str[i] == 'i' || str[i] == 'I' || str[i] == 'o' || str[i] == 'O'
|| str[i] == 'u' || str[i] == 'U') {
            count++;
        }
        i++;
    }
    return count;
}
void read_string(char *str, int maxlen) {
    int i = 0;
    while(((str[i] = getchar()) != EOF && str[i] != '\n') && i <
maxlen-1)
        i++;
    str[i] = '\0';
    return;
}

int main() {
    char str[64];
    printf("Enter string: ");
    read_string(str, 64);
    printf("No. of vowels = %d\n", count_vowels(str));
    return 0;
}
```

Output:



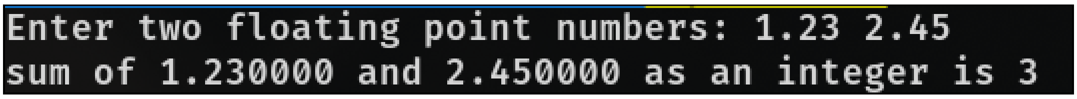
```
Enter string: Hello World
No. of vowels = 3
```

4. Write a program to read two floating point numbers. Add these numbers and assign the result to an integer. Finally, display the value of all the three variables.

Code:

```
#include <stdio.h>

int main() {
    float f1, f2;
    int sum;
    printf("Enter two floating point numbers: ");
    scanf("%f%f", &f1, &f2);

    sum = f1 + f2;
    printf("sum of %f and %f as an integer is %d\n", f1, f2, sum);
    return 0;
}
```

Output:



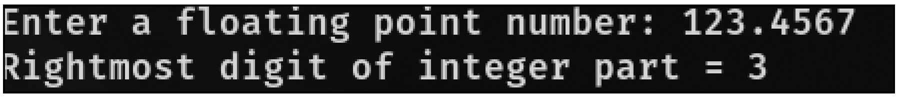
```
Enter two floating point numbers: 1.23 2.45
sum of 1.230000 and 2.450000 as an integer is 3
```

5. Write a program to read a floating point number. Display the rightmost digit of the integral part of the number

Code:

```
#include <stdio.h>
int main() {
    float f;
    int i;
    printf("Enter a floating point number: ");
    scanf("%f", &f);
    i = (int) f;
    printf("Rightmost digit of integer part = %d\n", i%10);
    return 0;
}
```

Output:



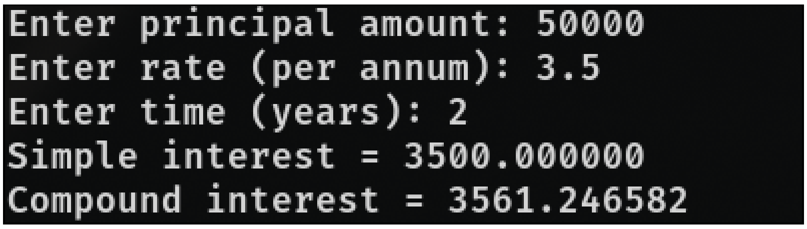
```
Enter a floating point number: 123.4567
Rightmost digit of integer part = 3
```

6. Write a program to calculate simple interest and compound interest.

Code:

```
#include <stdio.h>
#include <math.h>
float simple_interest(float p, float n, float r) {
    return (p * n * r) / 100;
}

float compound_interest(float p, float n, float r) {
    float result = p * pow((100 + r) / 100, n) - p;
    return result;
}

int main() {
    float p,r,t;

    printf("Enter principal amount: ");
    scanf("%f", &p);

    printf("Enter rate (per annum): ");
    scanf("%f", &r);

    printf("Enter time (years): ");
    scanf("%f", &t);

    printf("Simple interest = %f\n", simple_interest(p,t,r));
    printf("Compound interest = %f\n", compound_interest(p,t,r));
    return 0;
}
```

Output:



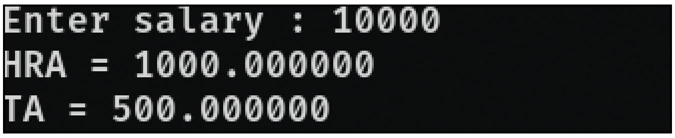
```
Enter principal amount: 50000
Enter rate (per annum): 3.5
Enter time (years): 2
Simple interest = 3500.000000
Compound interest = 3561.246582
```

7. Write a program to calculate salary of an employee given his basic pay (to be entered by the user), HRA = 10% of the basic pay, TA = 5% of basic pay. Define HRA and TA as constants and use them to calculate the salary of the employee.

Code:

```
#include <stdio.h>
#define HRA 10
#define TA 5

int main() {
    float salary;
    printf("Enter salary : ");
    scanf("%f", &salary);

    printf("HRA = %f\nTA = %f\n", (salary*HRA)/100, (salary*TA)/100);
    return 0;
}
```

output:



```
Enter salary : 10000
HRA = 1000.000000
TA = 500.000000
```

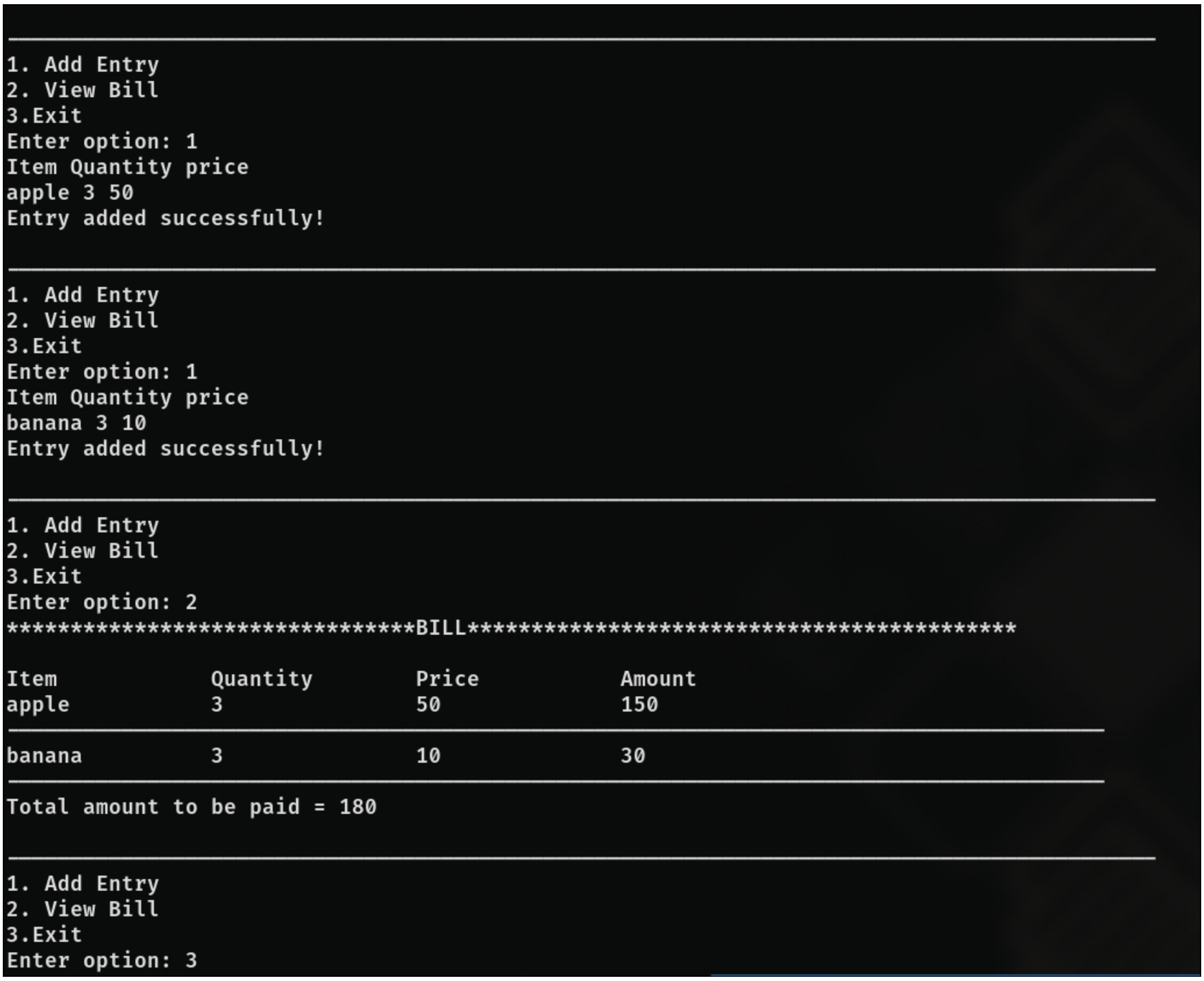
8. Write a program to prepare a grocery bill. Enter the name of the items purchased, quantity in which it is purchased, and its price per unit. Then display the bill in the following format:

code:

```
#include <stdio.h>
#include <stdlib.h>
#define ENTRIES 128
typedef struct {
    char item[64];
    int qty;
    int price;
    int amount;
} entry;
entry *bill[ENTRIES];
int bill_ptr = 0;
void init() {
    for (int i = 0; i < ENTRIES; i++) {
        bill[i] = NULL;
    }
    return;
}
entry *read_entry() {
    entry *e = (entry *)malloc(sizeof(entry));
    printf("Item Quantity price\n");
    scanf("%s%d%d", e->item, &e->qty, &e->price);
    return e;
}
void add_entry(entry *e) {
    bill[bill_ptr] = e;
    bill_ptr++;
    return;
}
void print_bill() {
    int total_amount = 0;
    printf(

"********************************BILL**********************************
"
        "*********\n\n");
    printf("Item\t\tQuantity\tPrice\t\tAmount\n");
    for (int i = 0; i < bill_ptr; i++) {
        printf("%s\t\t", bill[i]->item);
        printf("%d\t\t", bill[i]->qty);
        printf("%d\t\t", bill[i]->price);
        printf("%d\t\t", bill[i]->price * bill[i]->qty);
        printf(
            "\
n----------------------------------------------------------------"
            "----------------------\n");
        total_amount += bill[i]->price * bill[i]->qty;
    }
    printf("Total amount to be paid = %d\n", total_amount);
    return;
}

void print_menu() {
    printf("\
n----------------------------------------------------------------------
--------------------\n");
    printf("1. Add Entry\n2. View Bill\n3.Exit\n");
    printf("Enter option: ");
}

void read_option(int option) {
    switch (option) {
        case 1: {
            add_entry(read_entry());
            printf("Entry added successfully!\n");
            break;
        }
        case 2: {
            print_bill();
            break;
        }
        case 3: {
            return;
        }
        default: {
            printf("Invalid option\n");
            break;
        }
    }
    return;
```

```
}

int main() {
    int option;
    while(1){
        print_menu();
        scanf("%d", &option);
        read_option(option);
        if(option == 3) {
            break;
        }
    }

    return 0;
}
```

Output:



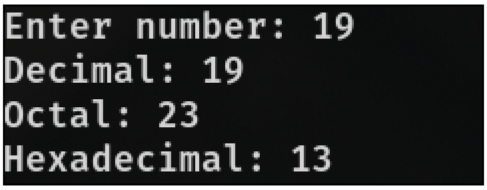
```
__________________________________________________________________________________________
1. Add Entry
2. View Bill
3.Exit
Enter option: 1
Item Quantity price
apple 3 50
Entry added successfully!

__________________________________________________________________________________________
1. Add Entry
2. View Bill
3.Exit
Enter option: 1
Item Quantity price
banana 3 10
Entry added successfully!

__________________________________________________________________________________________
1. Add Entry
2. View Bill
3.Exit
Enter option: 2
********************************BILL*******************************************

Item            Quantity        Price           Amount
apple           3               50              150
______________________________________________________________________________________
banana          3               10              30
______________________________________________________________________________________
Total amount to be paid = 180

__________________________________________________________________________________________
1. Add Entry
2. View Bill
3.Exit
Enter option: 3
```

9. Write a program to read an integer. Display the value of that integer in decimal, octal, and hexadecimal notation.

Code:

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>

void swap(char arr[], int i, int j) {
    char temp = arr[i];
    arr[i] = arr[j];
    arr[j] = temp;
    return;
}
void reverse(char *str) {
    for(int i = 0; i < strlen(str) / 2; i++){
        swap(str, i, strlen(str)-i-1);
    }
    return;
}

char *decimal_to_octal(int n) {
    int len=0;
    char str[128];

    while(n>0) {
        str[len++] = n % 8 + '0';
        n = n / 8;
    }
    str[len] = '\0';
    char *return_str = (char *) malloc(sizeof(char)*len);
    strcpy(return_str, str);
    reverse(return_str);
    return return_str;
}

char *decimal_to_hex(int n) {
    int len = 0;
    char str[128];
    char *return_str;
    char large_digits[] = { 'A', 'B', 'C', 'D', 'E', 'F'};
    while(n>0) {
        int remainder = n%16;
        str[len++] = remainder>9 ? large_digits[remainder-10]
: remainder + '0';
        n /= 16;
    }
    str[len] = '\0';
    return_str = (char *) malloc(sizeof(char)*len);
    strcpy(return_str, str);
    reverse(return_str);
    return return_str;
}

int main() {
    int n;
    char *octal, *hex;
    printf("Enter number: ");
    scanf("%d", &n);
    octal = decimal_to_octal(n);
    hex = decimal_to_hex(n);
    printf("Decimal: %d\nOctal: %s\nHexadecimal: %s\n", n,
octal, hex);
    return 0;
}
```

output:

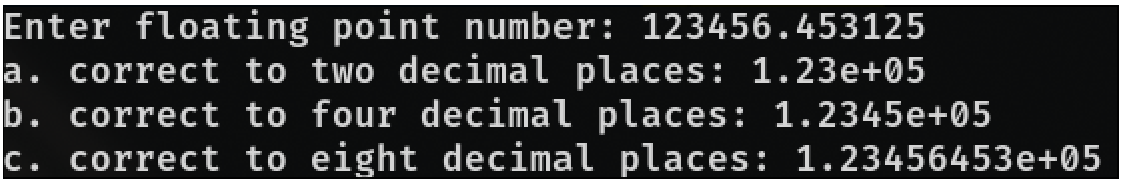


```
Enter number: 19
Decimal: 19
Octal: 23
Hexadecimal: 13
```

10. Write a program that prints a floating point value in exponential format with the
following specifications:
a. correct to two decimal places;
b. correct to four decimal places; and
c. correct to eight decimal places.
[Expected Output: value entered: 123456.453125 Value in exponent form: 1.234565e+05]
code:

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
int no_of_digits(int n) {
    int count=0;
    while(n) {
        n/=10;
        count++;
    }
    return count;
}

void reverse(char *str) {
    int len = strlen(str);
    for (int i = 0; i < len / 2; i++) {
        char temp = str[i];
        str[i] = str[len - i - 1];
        str[len - i - 1] = temp;
    }
}

char *exp_format(float f, int precision) {
    int exp=0, i=0, j=2;
    char str[64];
    char *return_str;

    float fraction = f - (int) f;
    fraction *= 10;
    while(fraction - (int) fraction) {
        str[i++] = ((int) fraction)%10 + '0';
        fraction *= 10;
    }
    str[i++] = ((int)fraction)%10 + '0';
    str[i] = '\0';
    reverse(str);
    while(f >= 10 || f <= -10) {
        str[i++] = ((int)f)%10 + '0';
        f = f/10;
        exp++;
    }
    i--;
    return_str = (char *) malloc((precision + 5 + no_of_digits(exp))
*sizeof(char)); // 2 (msb and '.') + precision + 4 (e, "+", ..., <no of
digits>) + null byte
    return_str[0] = ((int)f) % 10 + '0';
    return_str[1] = '.';
    while(precision) {
        return_str[j] = i >= 0 ? str[i] : '0';
        j++;
        i--;
        precision--;
    }
    return_str[j] = '\0';
    return_str[j++] = 'e';
    return_str[j++] = '+';
    if(exp<10)
        return_str[j++] = '0';
    while(exp>=10){
        return_str[j++] = exp/10 + '0';
        exp/=10;
    }
    return_str[j++] = exp + '0';
    return return_str;
}

int main() {
    float f;
    printf("Enter floating point number: ");
    scanf("%f", &f);
    printf("a. correct to two decimal places: %s\n", exp_format(f, 2));
    printf("b. correct to four decimal places: %s\n", exp_format(f,
4));
    printf("c. correct to eight decimal places: %s\n", exp_format(f,
8));
    return 0;
}
```

Output:



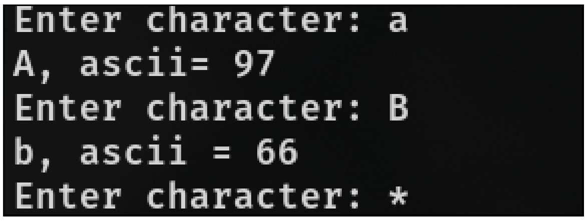
```
Enter floating point number: 123456.453125
a. correct to two decimal places: 1.23e+05
b. correct to four decimal places: 1.2345e+05
c. correct to eight decimal places: 1.23456453e+05
```

11. Write a program to read a character and print it. Also print its ASCII value. If the character is in lower case, print it in upper case and vice versa. Repeat the process until a '*' is entered

code:

```
#include <stdio.h>

int main() {
    char c;
    printf("Enter character: ");
    while((c = getchar()) != '*') {
        if(c >= 'a' && c <='z') {
            printf("%c, ascii= %d\nEnter character: ", c - ('a' - 'A'),
c);
        }else if(c >= 'A' && c <= 'Z'){
            printf("%c, ascii = %d\nEnter character: ", c + ('a' -
'A'), c);
        }
    }

    return 0;
}
```

output:



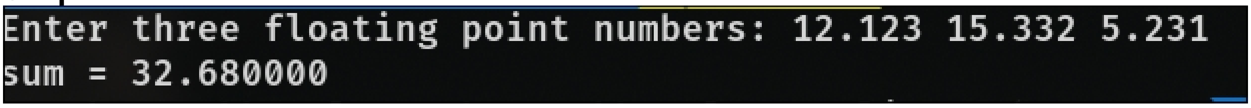
```
Enter character: a
A, ascii= 97
Enter character: B
b, ascii = 66
Enter character: *
```

12. Write a program to add three floating point numbers. The result should contain only two digits after the decimal.

Code:

```
#include <stdio.h>

int main() {
    float f1, f2, f3, sum;

    printf("Enter three floating point numbers: ");
    scanf("%f%f%f", &f1, &f2, &f3);

    sum = f1 + f2 + f3;
    sum *= 100;
    sum = (int)sum;
    sum = sum / 100;

    printf("sum = %f\n", sum);
    return 0;
}
```

Output:



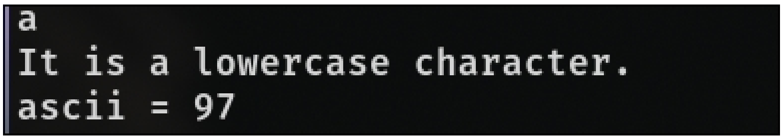
```
Enter three floating point numbers: 12.123 15.332 5.231
sum = 32.680000
```

13. Write a program to take input from the user and then check whether it is a number or a character. If it is a character, determine whether it is in upper case or lower case. Also print its ASCII value.

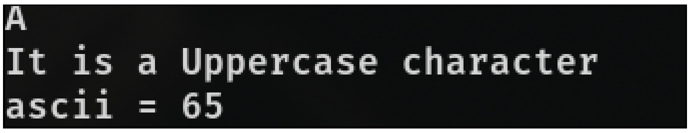
Code:

```
#include <stdio.h>

int is_number(char c){
    return c >= '0' && c <= '9';
}

int is_char(char c) {
    return (c >= 'a' && c <= 'z') || (c >= 'A' && c <= 'Z');
}

int is_upperCase(char c) {
    return c >= 'A' && c <= 'Z';
}

int main() {
    char c;
    c = getchar();
    if(is_number(c)) {
        printf("It is a number\n");
    }else if(is_char(c)) {
        is_upperCase(c) ?
        printf("It is a Uppercase character\n") :
        printf("It is a lowercase character.\n");

        printf("ascii = %d\n", c);
    }else{
        printf("Invalid input!\n");
    }
    return 0;}
```

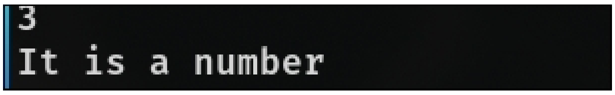
output:



```
a
It is a lowercase character.
ascii = 97
```



```
A
It is a Uppercase character
ascii = 65
```

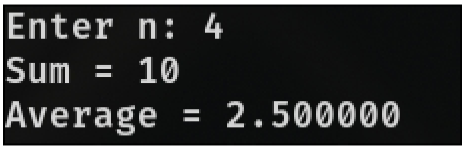


```
3
It is a number
```

14. Write a program to display sum and average of numbers from 1 to n. Use for loop.
Code:

```
#include <stdio.h>

int sum(int n) {
    int sum = 0;
    for(int i = 1; i <= n; i++)
        sum+=i;
    return sum;
}

float average(int n) {
    return ((float)sum(n) / (float)n);
}

int main() {
    int n;
    printf("Enter n: ");
    scanf("%d", &n);
    printf("Sum = %d\nAverage = %f\n", sum(n), average(n));
    return 0;
}
```

output:



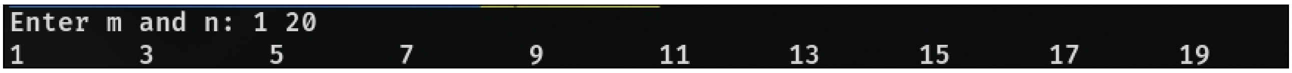
```
Enter n: 4
Sum = 10
Average = 2.500000
```

15. Write a program to print all odd numbers from m to n

CODE:

```
#include <stdio.h>

void print_odd(int m, int n) {
    int large, small;
    if(m>n) {
        large = m;
        small = n;
    }else {
        large = n;
        small = m;
    }
    for(int i = small; i <= large;i++) {
        if(i%2){
            printf("%d\t", i);
        }
    }
    return;
}

int main() {
    int m, n;
    printf("Enter m and n: ");
    scanf("%d%d", &m, &n);
    print_odd(m, n);
    return
}
```

output:

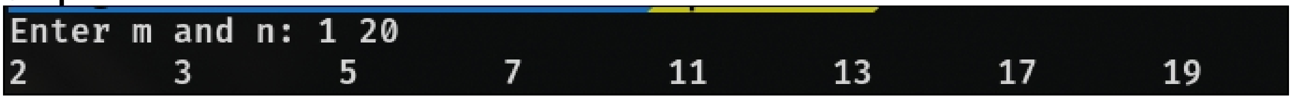


```
Enter m and n: 1 20
1       3       5       7       9       11      13      15      17      19
```

16. Write a program to print all prime numbers from m to n.
code:

```
#include <stdio.h>

int is_prime(int n) {
    if(n == 1)
        return 0;
    for(int i = 2; i < n; i++) {
        if((n%i) == 0)
            return 0;
    }
    return n;
}

void print_prime(int m, int n) {
    int large, small;
    if(m>n) {
        large = m;
        small = n;
    }else {
        large = n;
        small = m;
    }
    for(int i = small; i <= large; i++) {
        if(is_prime(i)) {
            printf("%d\t", i);
        }
    }
    return;
}

int main() {
    int m, n;
    printf("Enter m and n: ");
    scanf("%d%d", &m, &n);
    print_prime(m, n);
    return 0;
}
```

output:

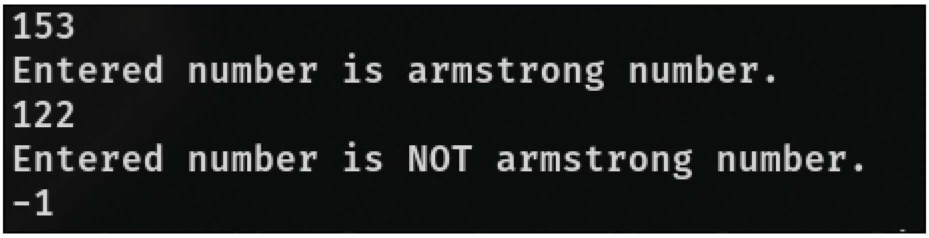


```
Enter m and n: 1 20
2       3       5       7       11      13      17      19
```

17. Write a program to read numbers until –1 is entered and display whether it is an Armstrong number or not.
Code:

```
#include <stdio.h>

int is_armstrong(int n) {
    int cube_sum = 0, total = n;
    while(n) {
        int r = n%10;
        cube_sum += r*r*r;
        n/=10;
    }
    return cube_sum == total;
}

int main() {
    int n;
    scanf("%d", &n);

    while(n != -1) {
        if(is_armstrong(n)){
            printf("Entered number is armstrong number.\n");
        }else {
            printf("Entered number is NOT armstrong number.\n");
        }
        scanf("%d", &n);
    }

    return 0;
}
```

output:



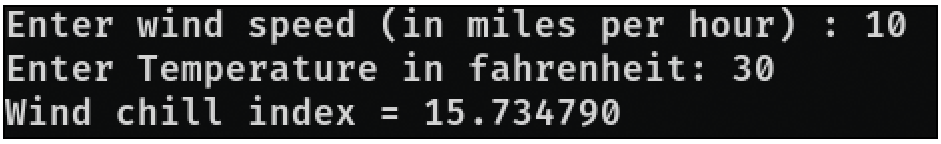
```
153
Entered number is armstrong number.
122
Entered number is NOT armstrong number.
-1
```

18. The wind chill index (WCI) is calculated from the wind speed v in miles per hour

and the temperature t in Fahrenheit. Three formulas are used, depending on the wind speed:
if (0 <= v <= 4) then WCI = t
if (v >=45) then WCI = 1.6t - 55
otherwise, WCI = 91.4 + (91.4 - t)(0.0203v - 0.304(v)1/2 - 0.474).
Write a program that can calculate the wind chill index.
Code:

```
#include <stdio.h>
#include <math.h>

float calc_wci(float v, float t) {
    if(v >= 0 && v <=4) {
        return t;
    }
    else if(v >= 45){
        return 1.6 * t - 55;
    }
    else {
        float result = 91.4 + (91.4 - t) * (0.0203 * v - 0.304 *
sqrt(v) - 0.474);
        return result;
    }
}

int main() {
    float v,t;
    printf("Enter wind speed (in miles per hour) : ");
    scanf("%f", &v);
    printf("Enter Temperature in fahrenheit: ");
    scanf("%f", &t);

    printf("Wind chill index = %f\n", calc_wci(v, t));
    return 0;
}
```

output:



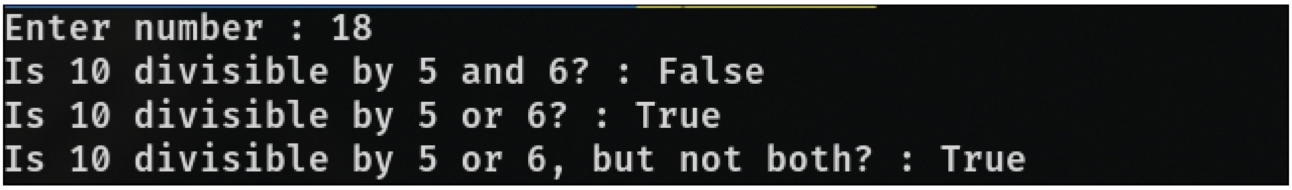
```
Enter wind speed (in miles per hour) : 10
Enter Temperature in fahrenheit: 30
Wind chill index = 15.734790
```

19. Write a program that asks the user to enter an integer and determines whether it is divisible by 5 and 6, whether it is divisible by 5 or 6, and whether it is divisible

by 5 or 6 but not both. For example, if your input is 10, the output should be:
Is 10 divisible by 5 and 6? false
Is 10 divisible by 5 or 6? true
Is 10 divisible by 5 or 6, but not both? True
code:

```
#include <stdio.h>

int is_divisible_by_n(int number, int n) { return !(number % n); }

// Logical Exclusive or
int xor (int a, int b) { return (a && !b) || (!a && b); }

  int main() {
  int number;
  printf("Enter number : ");
  scanf("%d", &number);

  printf("Is 10 divisible by 5 and 6? : %s\n",
      is_divisible_by_n(number, 5) && is_divisible_by_n(number, 6)
        ? "True"
        : "False");
  printf("Is 10 divisible by 5 or 6? : %s\n",
      is_divisible_by_n(number, 5) || is_divisible_by_n(number, 6)
        ? "True"
        : "False");
  printf("Is 10 divisible by 5 or 6, but not both? : %s\n",
      xor(is_divisible_by_n(number, 5), is_divisible_by_n(number, 6))
        ? "True"
        : "False");

  return 0;
}
```

output:



```
Enter number : 18
Is 10 divisible by 5 and 6? : False
Is 10 divisible by 5 or 6? : True
Is 10 divisible by 5 or 6, but not both? : True
```

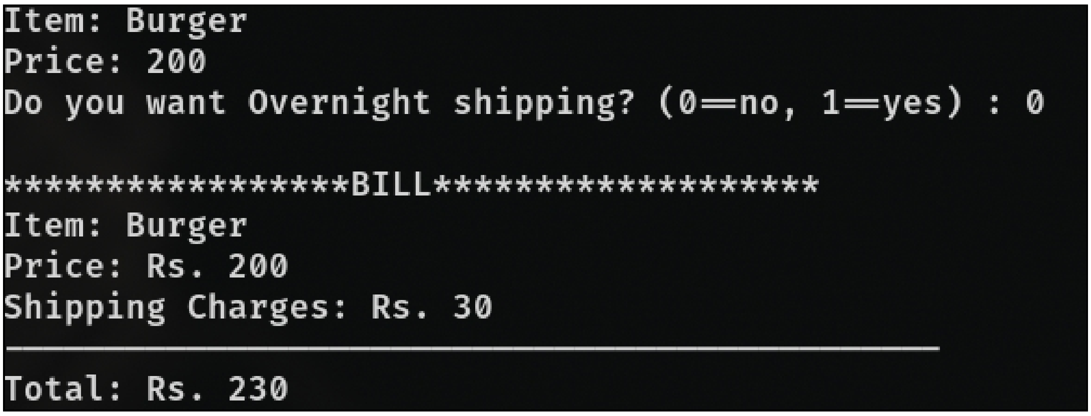
20. McDonald’s wants you to write a program to take orders from the Internet. Your program asks for the item, its price, and if overnight shipping is wanted. Regular shipping for items under Rs.100 is Rs20.00; for items Rs100 or more shipping is

Rs30.00. For overnight delivery add Rs50.00. For example, the output might be: Enter the item: Burger Enter the price: 450 Overnight delivery (0==no, 1==yes): 1 Invoice: Burger Rs.450 shipping Rs.80 total Rs. 530.

code:

```
#include <stdio.h>

void generate_invoice(int price, char item[], int overnight_shipping) {
    int shipping;
    if(price < 100)
        shipping = 20;
    else
        shipping = 30;
    if(overnight_shipping)
        shipping += 50;

    printf("\n*****************BILL*******************\n");
    printf("Item: %s\n", item);
    printf("Price: Rs. %d\n", price);
    printf("Shipping Charges: Rs. %d\n", shipping);
    printf("----------------------------------------------\n");
    printf("Total: Rs. %d\n", price + shipping);
    return;
}

int main() {
    int price, overnight_shipping;
    char item[64];

    printf("Item: ");
    scanf("%s", item);
    printf("Price: ");
    scanf("%d", &price);
    printf("Do you want Overnight shipping? (0==no, 1==yes) : ");
    scanf("%d", &overnight_shipping);

    generate_invoice(price, item, overnight_shipping);
    return 0;
}
```

output:



```
Item: Burger
Price: 200
Do you want Overnight shipping? (0==no, 1==yes) : 0

*****************BILL*******************
Item: Burger
Price: Rs. 200
Shipping Charges: Rs. 30
----------------------------------------------
Total: Rs. 230
```