

*****Program 0.c *****

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

```
int main() {
```

```
    int i; //4 bytes
```

```
    float f; //4 bytes
```

```
    char c; // 1 byte
```

```
    long int li;
```

```
    float lf;
```

```
    printf("Size of int is %zu bytes\n",sizeof(int));
```

```
    printf("Size of float is %zu bytes\n",sizeof(float));
```

```
    printf("Size of char is %zu bytes\n",sizeof(char));
```

```
    printf("Size of double is %zu bytes\n",sizeof(double));
```

```
    printf("Size of long int is %zu bytes\n",sizeof(long int));
```

```
    printf("Size of short int is %zu bytes\n",sizeof(short));
```

```
    printf("Size of signed int is %zu bytes\n",sizeof(signed int));
```

```
    printf("Size of unsigned int is %zu bytes\n",sizeof(unsigned int));
```

```
    return 2;
```

```
}
```

*****Program 1.c *****

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

```
int main() {
```

```
    int x;
```

```
    float m = 0, s = 0;
```

```
    int N = 5;
```

```
    int c = 0;
```

```
    while (c < N) {
```

```
        scanf("%d", &x);
```

```
        s = s + x;
```

```
        c = c + 1;
```

```
    }
```

```
    m = s / N;
```

```
    printf("Sum is %f - Mean value = %f\n", s, m);
```

```
}
```

*****Program 2.c *****

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

```
int main() {
```

```
    int x;
```

```
    float m = 0, s = 0, sx = 0, var = 0;
```

```
    int N = 5;
```

```
    int c = 0;
```

```
    while (c < N) {
```

```
        scanf("%d", &x);
```

```
        s = s + x;
```

```
        sx = sx + (x * x);
```

```
        c = c + 1;
```

```
    }
```

```
    m = s / N;
```

```
    var = sx / N - (m * m);
```

```
    printf("Sum is %f - Mean value = %f, Var = %f\n", s, m, var);
```

```
    //printf("Sum is %.2f - Mean value = %.2f, Var = %.2f\n", s, m, var);
```

```
}
```

*****Program 3.c *****

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

```
int main() {
```

```
    float r;
```

```
    r = 10/3;
```

```
    printf("10/3 = %f\n",r);
```

```
    r = 10.0/3;
```

```
    printf("10.0/3 = %f\n",r);
```

```
    r = 10/3.0;
```

```
    printf("10/3.0 = %f\n",r);
```

```
    r = (float) 10/3;
```

```
    printf("(float) 10/3 = %f\n",r);
```

```
    r = (int) (10.0/3);
```

```
    printf("(int) (10.0/3) = %f\n",r);
```

```
char ch = 'a';
printf("'a' is %c its integer value is %d\n",ch, (int) ch);
ch = ch + 1;
printf("now ch is %c and its integer value is %d\n", ch, ch);
```

```
float exp;
int a = 10, b = 3, c = 5;
```

```
exp = a + b + c / 2.0;
printf("10 + 3 + 5 / 2.0 = %f\n", exp);
```

```
exp = (a + b + c) / 2.0;
printf("(10 + 3 + 5) / 2.0 = %f\n", exp);
```

```
exp = a / c * b ;
printf("10 / 5 * 3 = %f\n", exp);
```

```
exp = (float) 10 / (5 * 3) ;
printf("(float) 10 / (5 * 3) = %f\n", exp);
printf("truncated value upto 2 decimals only %.2f\n", exp);
```

```
}
*****Program 4.c *****
```

```
#include<stdio.h>
int main() {
    int n;
    scanf("%d",&n);
    int temp=n;
    int r=0;
    // Reverse the number and store it in r
    while(temp) {
        r = r*10 + temp%10;
        temp = temp/10;
    }
    if ( r == n) {
        printf("Yes\n");
    }
    else {
        printf("No\n");
    }
}
```

*****Program 5.c *****

```
#include<stdio.h>
int main() {
    char ch;
    int open=0;
    scanf("%c",&ch);
    //printf("what is read: %c\n",ch);
    while(ch != '\n') {
        if(ch=='/' || ch=='*') {
            printf("%c(",ch);
            open++;
        }
        else {
            printf("%c",ch);
        }
        scanf("%c",&ch);
        //printf("what is read: %c\n",ch);
    }
    int i;
    for(i = 0; i < open; i++){
        printf(")");
    }
    printf("\n");
    return 0;
}
```

*****Program 6.c *****

```
//Read a vector of d-dimension
//sum its positive components and negative components
#include<stdio.h>
int main() {
    int d;
    int i;
    float x;
    float ps = 0;
    int pc = 0;
    float ns = 0;
    int nc = 0;

    scanf("%d",&d);
    for (i = 0; i < d; i++) {
        scanf("%f",&x);
        if (x >= 0) {
```

```

        pc++;
        ps = ps + x;
    } else {
        nc++;
        ns = ns + x;
    }
}
printf("%d %.2f %d %.2f\n",pc, ps, nc, ns);
}
*****Program 6.1.c *****
//read a vector of d-dimension
//sum its positive and negative components
//repeat it for n test cases.
#include<stdio.h>
int main() {
    int d;
    int i;
    float x;
    float ps = 0;
    int pc = 0;
    float ns = 0;
    int nc = 0;

    int n;
    scanf("%d",&n);
    while (n > 0) { //loop for running n test-cases.
        scanf("%d",&d);
        pc = 0;
        ps = 0;
        nc = 0;
        ns = 0;
        for (i = 0; i < d; i++) {
            scanf("%f",&x);
            if (x >= 0) {
                pc++;
                ps = ps + x;
            } else {
                nc++;
                ns = ns + x;
            }
        }
    }
    printf("%d %.2f %d %.2f\n",pc, ps, nc, ns);
}

```

```

    n--;
}
}
*****Program 7.c *****
//compute sine series for n terms..
//sin(x) = x - x^3/3! + x^5/5! - x^7/7!+...+(-1)^n-1 x^(2n-1)/(2n-1)!
//ith term = (-1)^(i-1) x^(2i-1)/(2i-1)!
//(i+1)th term = (-1)^i x^(2i+1)/(2i+1)!
//recursive way (-1) x^2/(2i * (2i+1)) * ith term
#include<stdio.h>
int main() {
    float x;
    int n;
    int i;
    float sinx = 0;
    float term = 0;

    scanf("%f",&x);
    scanf("%d",&n);
    sinx = x;
    term = x;
    for (i = 1; i < n; i++) {
        term = - (x * x * term) / ((2*i) * (2*i+1));
        sinx = sinx + term;
    }
    printf("sin(%.2f) is %f\n", x, sinx);
}

```

```

*****Program 8.c *****
//decimal (positive integer) to binary..
#include<stdio.h>
#include<math.h>
int main() {
    int w;
    int i = 0;
    int v = 0;
    scanf("%d",&w);
    while (w != 0) {
        v = (w % 2) * (int) pow(10,i) + v;
        w = w / 2;
        i = i + 1;
    }
}

```

```

    }
    printf("%d\n", v);
}
*****Program 9.c *****
//float to binary..
//231.45 = 2*10^2 + 3*10^1 + 1*10^0 + 4*10^(-1) + 5*10^(-2)
//101.011 = 1*2^2 + 0*2^1 + 1*2^0 + 0*2^(-1) + 1*2^(-2) + 1*2^(-3)
//      = 5.375
//convert 5 to binary by division of 2 and .375 to binary by dividing with 2^(-1)

#include<stdio.h>
#include<math.h>
int main() {
    int w;
    int i = 0;
    int v = 0;

    float x;
    float r;
    int t;

    scanf("%f",&x);
    w = (int) x;
    r = x - w;
    printf("%d %f\n",w,r);

    while (w != 0) {
        v = (w % 2) * (int) pow(10,i) + v;
        w = w / 2;
        i = i + 1;
    }
    printf("%d.", v);

    while (r > 0) {
        r = r * 2;
        t = (int) r;
        r = r - t;
        printf("%d",t);
    }
    printf("\n");
}

```

*****Program 10.c *****

```
//iterative method
//f(x) = x^3 - 2.5 x^2 - 2.46 x + 3.96 = 0;
//example input -4 4 1
#include<stdio.h>
int main() {
    double min, max, delta;
    double x, y;
    scanf("%lf%lf%lf", &min, &max, &delta);
    x = min;
    printf("x\tf(x)\n");
    while (x <= max) {
        y = (x * x * x) - (2.5 * x * x) - (2.46 * x) + 3.96;
        printf("%.4f\t%.4f\n", x, y);
        x = x + delta;
    }
}
```

*****Program 11.1.c *****

```
//Bisection method
#include<stdio.h>
#include<stdlib.h>
#include<math.h>
int main() {
    double x0, x1, x2, e;
    double y0, y1, y2;
    double d = 25;
    int i;

    scanf("%lf%lf%lf", &x0, &x1, &e);
    y0 = x0 * x0 - d;
    y1 = x1 * x1 - d;
    i = 0;
    if (y0 * y1 > 0) { //if same sign
        printf("Starting values %lf %lf not suitable\n", x0, x1);
        exit(1);
    }

    while (fabs((x1 - x0) ) > e) {
        //change the above to (x1-x0)/x1 for small roots of order 10^-6
        x2 = (x0 + x1) / 2.0;
        y2 = x2 * x2 - d;
        i = i + 1;
    }
}
```



```

printf("%d\t%.4f\t%.4f\t%.4f\t%.4f\t",i,x0,x1,x2,y2);
if (y0 * y2 > 0) {
    x0 = x2;
    printf("x0 <- x2\n");
} else {
    x1 = x2;
    printf("x1 <- x2\n");
}
}
printf("number of iterations = %d\n",i);
printf("f(%.4f) = %.4f\n", x2, y2);

}

```

*****Program 12.c *****

```

//Newton-Raphson method
#include<stdio.h>
#include<math.h>
int main() {
    double x0, x1, eps, delta;
    int n, i;
    float d = 25;
    double f0, df0;
    int flag = 0;

    scanf("%lf%lf%lf%d",&x0, &eps, &delta, &n);

    for (i = 0; i < n; i++) { //repeat for so many iterations.
        f0 = x0 * x0 - d;
        df0 = 2 * x0;
        if (fabs(df0) <= delta) {
            printf("slope is too small %.4f\n", df0);
        }

        x1 = x0 - f0/df0;
        if (fabs (x1 - x0) < eps) {
            //change to (x1-x0)/x1 for small roots of order 10^-6
            flag = 1;
            break; //exit from the for-loop;
        }
        x0 = x1;
    }
}

```

```
if (flag == 1) {  
    f0 = x1 * x1 - d;  
    printf("Convergent solution root = %.4lf, f(%.4lf) = %.4lf, it = %d\n", x1, x1, f0, i);  
} else {  
    printf("Does not converge in %d iterations\n", n);  
}  
  
}
```