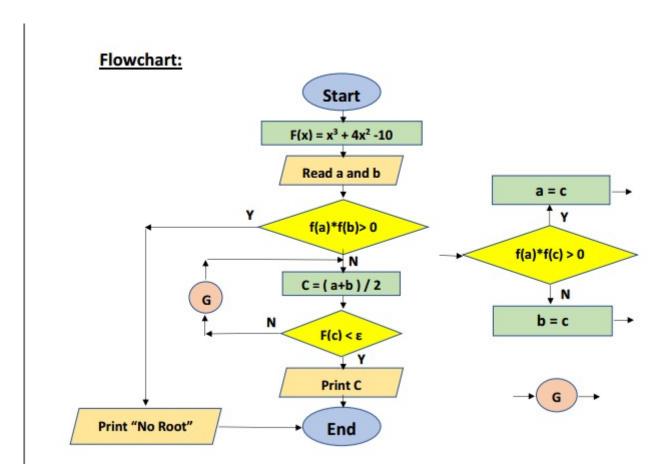
## **Bisection Method**

## Algorithm:

- 1. Set the function or the equation (f(x) = ax2 + bx + c)
- 2. Set the accepting error limit (  $\epsilon$  ).
- 3. Take input the limits (a, b).
- 4. If f(a) \* f(b) is greater than 0, go to 5; else go to 6.
- 5. Print "There is no root within this limit, exit.
- 6. Repeat 7-9 till f(c) is less than error limit (  $\epsilon$  ).
- 7. Make c as the average of a & b.
- 8. If f(c)\*f(a) < 0, set b=c.
- 9. If f(c)\*f(b) < 0, set a = c.
- 10. Print c as the root of the equation; then exit.



## Source code:

```
#include<stdio.h>
#include<string.h>
#include<stdlib.h>
#include<math.h>
#define EPS 10.0e-6
#define f(x) (x^*x^*x+4^*x^*x-10)
int main(){
  int n=100,i;
  double a=1.25,b=1.5,c;
  if(f(a)*f(b)>0)exit(0);
  printf("-----
                                                                                          --\n");
  printf("ITER
                                           f(a)
                                                       f(b)
                                                                   f(c)\n");
                         b
                                  С
printf("-----
  for(int i=1;i \le n;i++){
    c=(a+b)/2;
                                                    f^n,i,a,b,c,f(a),f(b),f(c);
    printf("%d
                  %lf
                        %lf
                              %lf
                                    %lf
                                            %lf
    if(fabs(f(c)) < EPS){
       printf("root-%lf",c);
      break;
    }
    if(f(c)*f(a)<0){
      b=c;
    }
    else a=c;
  }
  return 0;
}
```

## **Output:**

ITER	a	b	С	f(a)	f(b)	f(c)
1	1.250000	1.500000	1.375000	-1.796875	2.375000	0.162109
2	1.250000	1.375000	1.312500	-1.796875	0.162109	-0.848389
3	1.312500	1.375000	1.343750	-0.848389	0.162109	-0.350983
4	1.343750	1.375000	1.359375	-0.350983	0.162109	-0.096409
5	1.359375	1.375000	1.367188	-0.096409	0.162109	0.032356
6	1.359375	1.367188	1.363281	-0.096409	0.032356	-0.032150
7	1.363281	1.367188	1.365234	-0.032150	0.032356	0.000072
8	1.363281	1.365234	1.364258	-0.032150	0.000072	-0.016047
9	1.364258	1.365234	1.364746	-0.016047	0.000072	-0.007989
10	1.364746	1.365234	1.364990	-0.007989	0.000072	-0.003959
11	1.364990	1.365234	1.365112	-0.003959	0.000072	-0.001944
12	1.365112	1.365234	1.365173	-0.001944	0.000072	-0.000936
13	1.365173	1.365234	1.365204	-0.000936	0.000072	-0.000432
14	1.365204	1.365234	1.365219	-0.000432	0.000072	-0.000180
15	1.365219	1.365234	1.365227	-0.000180	0.000072	-0.000054
16	1.365227	1.365234	1.365231	-0.000054	0.000072	0.000009
root-1.365231						