# CODE FOR THE FIRST QUESTION

```c
#include <stdio.h>

int main(){

float a, b, bek, dny, err, totbek = 0, totdny = 0, totper = 0;
int cnt = 0;
char sym;

printf("Enter an operation (Example: 3+2): ");          /* Takes the input from the user */
scanf("%f%c%f", &a, &sym, &b);

if (sym == '+')
{
    bek = a + b;                                        /* Calculates the expected result */
    printf("\nExpected Result: %.2f\n", bek);
    totbek += bek;                                      /* Adds the expected result to the total expected results */

    dny = a - b;                                        /* Calculates the experimental result */
    printf("\nExperimental Result: %.2f\n", dny);
    totdny += dny;

    err = bek - dny;
    if (err < 0)                                        /* Takes the absolute value of the error */
    {
        err *= -1;
    }
    printf("\nError: %.2f\n", err);

    if (bek == 0){                          /* If the expected equals 0, program gives error to prevent division by zero */
        totper += 0;
        cnt += 1;                           /* Increments the counter for how many times expected equals error */
        printf("\nCouldn't calculate the error percentage due to division by zero mistake!\n");
    }
    else {
        totper += err/bek*100;              /* Adds the value of the error to the total error percentage */
        printf("\nError Percentage: %.2f\n", (err/bek*100));
    }

}

else if (sym == '-')
{
    bek = a - b;
    printf("\nExpected Result: %.2f\n", bek);
```

```
45      totbek += bek;
46
47      dny = a * b;
48      printf("\nExperimental Result: %.2f\n", dny);
49      totdny += dny;
50
51      err = bek - dny;
52      if (err < 0)
53      {
54          err *= -1;
55      }
56      printf("\nError: %.2f\n", err);
57      if (bek == 0){
58          totper += 0;
59          cnt += 1;
60          printf("\nCouldn't calculate the error percentage due to division by
    zero mistake!\n");
61      }
62      else {
63          totper += err/bek*100;
64          printf("\nError Percentage: %.2f\n", (err/bek*100));
65      }
66  }
67
68  else if (sym == '*')
69  {
70      bek = a * b;
71      printf("\nExpected Result: %.2f\n", bek);
72      totbek += bek;
73
74      dny = a + b;
75      printf("\nExperimental Result: %.2f\n", dny);
76      totdny += dny;
77
78      err = bek - dny;
79      if (err < 0)
80      {
81          err *= -1;
82      }
83      printf("\nError: %.2f\n", err);
84      if (bek == 0){
85          totper += 0;
86          cnt += 1;
87          printf("\nCouldn't calculate the error percentage due to division by
    zero mistake!\n");
88      }
89      else {
90          totper += err/bek*100;
91          printf("\nError Percentage: %.2f\n", (err/bek*100));
92      }
93  }
94
95  else if (sym == '/')
96  {
97      bek = a / b;
```

```
    printf("\nExpected Result: %.2f\n", bek);
    totbek += bek;

    dny = a - b;
    printf("\nExperimental Result: %.2f\n", dny);
    totdny += dny;

    err = bek - dny;
    if (err < 0)
    {
        err *= -1;
    }
    printf("\nError: %.2f\n", err);
    if (bek == 0){
        totper += 0;
        cnt += 1;
        printf("\nCouldn't calculate the error percentage due to division by zero mistake!\n");
    }
    else {
        totper += err/bek*100;
        printf("\nError Percentage: %.2f\n", (err/bek*100));
    }
}

printf("\nEnter an operation (Example: 3+2): ");
scanf("%f%c%f", &a, &sym, &b);

if (sym == '+')
{
    bek = a + b;
    printf("\nExpected Result: %.2f\n", bek);
    totbek += bek;

    dny = a - b;
    printf("\nExperimental Result: %.2f\n", dny);
    totdny += dny;

    err = bek - dny;
    if (err < 0)
    {
        err *= -1;
    }
    printf("\nError: %.2f\n", err);

    if (bek == 0){
        totper += 0;
        cnt += 1;
        printf("\nCouldn't calculate the error percentage due to division by zero mistake!\n");
    }
    else {
        totper += err/bek*100;
        printf("\nError Percentage: %.2f\n", (err/bek*100));
    }
```

```
151
152 }
153
154 else if (sym == '-')
155 {
156     bek = a - b;
157     printf("\nExpected Result: %.2f\n", bek);
158     totbek += bek;
159
160     dny = a * b;
161     printf("\nExperimental Result: %.2f\n", dny);
162     totdny += dny;
163
164     err = bek - dny;
165     if (err < 0)
166     {
167         err *= -1;
168     }
169     printf("\nError: %.2f\n", err);
170     if (bek == 0){
171         totper += 0;
172         cnt += 1;
173         printf("\nCouldn't calculate the error percentage due to division by
    zero mistake!\n");
174     }
175     else {
176         totper += err/bek*100;
177         printf("\nError Percentage: %.2f\n", (err/bek*100));
178     }
179 }
180
181 else if (sym == '*')
182 {
183     bek = a * b;
184     printf("\nExpected Result: %.2f\n", bek);
185     totbek += bek;
186
187     dny = a + b;
188     printf("\nExperimental Result: %.2f\n", dny);
189     totdny += dny;
190
191     err = bek - dny;
192     if (err < 0)
193     {
194         err *= -1;
195     }
196     printf("\nError: %.2f\n", err);
197     if (bek == 0){
198         totper += 0;
199         cnt += 1;
200         printf("\nCouldn't calculate the error percentage due to division by
    zero mistake!\n");
201     }
202     else {
203         totper += err/bek*100;
```

```
204             printf("\nError Percentage: %.2f\n", (err/bek*100));
205         }
206     }
207 
208     else if (sym == '/')
209     {
210         bek = a / b;
211         printf("\nExpected Result: %.2f\n", bek);
212         totbek += bek;
213 
214         dny = a - b;
215         printf("\nExperimental Result: %.2f\n", dny);
216         totdny += dny;
217 
218         err = bek - dny;
219         if (err < 0)
220         {
221             err *= -1;
222         }
223         printf("\nError: %.2f\n", err);
224         if (bek == 0){
225             totper += 0;
226             cnt += 1;
227             printf("\nCouldn't calculate the error percentage due to division by
    zero mistake!\n");
228         }
229         else {
230             totper += err/bek*100;
231             printf("\nError Percentage: %.2f\n", (err/bek*100));
232         }
233     }
234 
235     printf("\nEnter an operation (Example: 3+2): ");
236     scanf("%f%c%f", &a, &sym, &b);
237 
238     if (sym == '+')
239     {
240         bek = a + b;
241         printf("\nExpected Result: %.2f\n", bek);
242         totbek += bek;
243 
244         dny = a - b;
245         printf("\nExperimental Result: %.2f\n", dny);
246         totdny += dny;
247 
248         err = bek - dny;
249         if (err < 0)
250         {
251             err *= -1;
252         }
253         printf("\nError: %.2f\n", err);
254 
255         if (bek == 0){
256             totper += 0;
257             cnt += 1;
```

```
258         printf("\nCouldn't calculate the error percentage due to division by
    zero mistake!\n");
259     }
260     else {
261         totper += err/bek*100;
262         printf("\nError Percentage: %.2f\n", (err/bek*100));
263     }
264 
265 }
266 
267 else if (sym == '-')
268 {
269     bek = a - b;
270     printf("\nExpected Result: %.2f\n", bek);
271     totbek += bek;
272 
273     dny = a * b;
274     printf("\nExperimental Result: %.2f\n", dny);
275     totdny += dny;
276 
277     err = bek - dny;
278     if (err < 0)
279     {
280         err *= -1;
281     }
282     printf("\nError: %.2f\n", err);
283     if (bek == 0){
284         totper += 0;
285         cnt += 1;
286         printf("\nCouldn't calculate the error percentage due to division by
    zero mistake!\n");
287     }
288     else {
289         totper += err/bek*100;
290         printf("\nError Percentage: %.2f\n", (err/bek*100));
291     }
292 }
293 
294 else if (sym == '*')
295 {
296     bek = a * b;
297     printf("\nExpected Result: %.2f\n", bek);
298     totbek += bek;
299 
300     dny = a + b;
301     printf("\nExperimental Result: %.2f\n", dny);
302     totdny += dny;
303 
304     err = bek - dny;
305     if (err < 0)
306     {
307         err *= -1;
308     }
309     printf("\nError: %.2f\n", err);
310     if (bek == 0){
```

```
311         totper += 0;
312         cnt += 1;
313         printf("\nCouldn't calculate the error percentage due to division by
    zero mistake!\n");
314     }
315     else {
316         totper += err/bek*100;
317         printf("\nError Percentage: %.2f\n", (err/bek*100));
318     }
319 }
320 
321 else if (sym == '/')
322 {
323     bek = a / b;
324     printf("\nExpected Result: %.2f\n", bek);
325     totbek += bek;
326 
327     dny = a - b;
328     printf("\nExperimental Result: %.2f\n", dny);
329     totdny += dny;
330 
331     err = bek - dny;
332     if (err < 0)
333     {
334         err *= -1;
335     }
336     printf("\nError: %.2f\n", err);
337     if (bek == 0){
338         totper += 0;
339         cnt += 1;
340         printf("\nCouldn't calculate the error percentage due to division by
    zero mistake!\n");
341     }
342     else {
343         totper += err/bek*100;
344         printf("\nError Percentage: %.2f\n", (err/bek*100));
345     }
346 }
347 
348 printf("\nEnter an operation (Example: 3+2): ");
349 scanf("%f%c%f", &a, &sym, &b);
350 
351 if (sym == '+')
352 {
353     bek = a + b;
354     printf("\nExpected Result: %.2f\n", bek);
355     totbek += bek;
356 
357     dny = a - b;
358     printf("\nExperimental Result: %.2f\n", dny);
359     totdny += dny;
360 
361     err = bek - dny;
362     if (err < 0)
363     {
```

```
364         err *= -1;
365     }
366     printf("\nError: %.2f\n", err);
367 
368     if (bek == 0){
369         totper += 0;
370         cnt += 1;
371         printf("\nCouldn't calculate the error percentage due to division by
    zero mistake!\n");
372     }
373     else {
374         totper += err/bek*100;
375         printf("\nError Percentage: %.2f\n", (err/bek*100));
376     }
377 
378 }
379 
380 else if (sym == '-')
381 {
382     bek = a - b;
383     printf("\nExpected Result: %.2f\n", bek);
384     totbek += bek;
385 
386     dny = a * b;
387     printf("\nExperimental Result: %.2f\n", dny);
388     totdny += dny;
389 
390     err = bek - dny;
391     if (err < 0)
392     {
393         err *= -1;
394     }
395     printf("\nError: %.2f\n", err);
396     if (bek == 0){
397         totper += 0;
398         cnt += 1;
399         printf("\nCouldn't calculate the error percentage due to division by
    zero mistake!\n");
400     }
401     else {
402         totper += err/bek*100;
403         printf("\nError Percentage: %.2f\n", (err/bek*100));
404     }
405 }
406 
407 else if (sym == '*')
408 {
409     bek = a * b;
410     printf("\nExpected Result: %.2f\n", bek);
411     totbek += bek;
412 
413     dny = a + b;
414     printf("\nExperimental Result: %.2f\n", dny);
415     totdny += dny;
416 
```

```
417     err = bek - dny;
418     if (err < 0)
419     {
420         err *= -1;
421     }
422     printf("\nError: %.2f\n", err);
423     if (bek == 0){
424         totper += 0;
425         cnt += 1;
426         printf("\nCouldn't calculate the error percentage due to division by
    zero mistake!\n");
427     }
428     else {
429         totper += err/bek*100;
430         printf("\nError Percentage: %.2f\n", (err/bek*100));
431     }
432 }
433 
434 else if (sym == '/')
435 {
436     bek = a / b;
437     printf("\nExpected Result: %.2f\n", bek);
438     totbek += bek;
439 
440     dny = a - b;
441     printf("\nExperimental Result: %.2f\n", dny);
442     totdny += dny;
443 
444     err = bek - dny;
445     if (err < 0)
446     {
447         err *= -1;
448     }
449     printf("\nError: %.2f\n", err);
450     if (bek == 0){
451         totper += 0;
452         cnt += 1;
453         printf("\nCouldn't calculate the error percentage due to division by
    zero mistake!\n");
454     }
455     else {
456         totper += err/bek*100;
457         printf("\nError Percentage: %.2f\n", (err/bek*100));
458     }
459 }
460 
461 if (cnt == 4)                           /* If all the expected value are 0
    then the program gives this message */
462     printf("\nCouldn't calculate the aveargе error percentage due to division
    by zero mistake!\n");
463 else                                    /* The program calculates the average
    error by dividing total error percentages by the amount of operation that
    doesn't have the expected error value equal to 0 */
464     printf("\nAverage Error Value: %.2f%%\n", totper/(4-cnt));
465 
```

```
466   return 0;
467   }
```

# SCREENSHOT OF FIRST QUESTION'S OUTPUT

```
albay@albay-VirtualBox:~/Desktop$ gcc -ansi MuhammetFatih_Albayın1.c -o  m
albay@albay-VirtualBox:~/Desktop$ ./m
Enter an operation (Example: 3+2): 3+2

Expected Result: 5.00

Experimental Result: 1.00

Error: 4.00

Error Percentage: 80.00

Enter an operation (Example: 3+2): 6*4

Expected Result: 24.00

Experimental Result: 10.00

Error: 14.00

Error Percentage: 58.33

Enter an operation (Example: 3+2): 8-5

Expected Result: 3.00

Experimental Result: 40.00

Error: 37.00

Error Percentage: 1233.33

Enter an operation (Example: 3+2): 9/3

Expected Result: 3.00

Experimental Result: 6.00

Error: 3.00

Error Percentage: 100.00

Average Error Value: 367.92%
albay@albay-VirtualBox:~/Desktop$
```

# CODE FOR THE SECOND QUESTION

```
1   MuhammetFatih_Albayın2.txt
2   #include <stdio.h>
3
4   int main(){
5
6   float cap, rad;
7
8   printf("Enter the diameter of a circle: ");            /* Takes the diameter
    as an input */
9   scanf("%f", &cap);
10  rad = cap / 2;                                         /* Calculates the
    radius */
11
12  printf("r = %.1f", rad);
13  printf("\nArea of sphere: %.2f\n", (3.14*4*rad*rad));   /* Calculates the area
    using the formula */
14
15  return 0;
16
17  }
```

# SCREENSHOT OF SECOND QUESTION'S OUTPUT

```
albay@albay-VirtualBox:~/Desktop$ gcc -ansi MuhammetFatih_Albayın2.c -o  f
albay@albay-VirtualBox:~/Desktop$ ./f
Enter the diameter of a circle: 2
r = 1.0
Area of sphere: 12.56
albay@albay-VirtualBox:~/Desktop$
```