1. #include "stdio.h"
2. #include "string.h"
3. #include "memory.h"
4. #define MAXLEN 100//定义大数的位数
6. **struct** Bigint{
7. **int** len;
8. **int** digits[MAXLEN];
9. };//定义大数结构体
11. //////////////////
12. **void** format()
13. {
14. printf("\t\t  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");
15. printf("\t\t  -----------1-大数加法-------------------\n");
16. printf("\t\t  -----------2-大数减法-------------------\n");
17. printf("\t\t  -----------3-大数乘法-------------------\n");
18. printf("\t\t  -----------4-大数除法-------------------\n");
19. printf("\t\t  -----------5-程序退出-------------------\n");
20. printf("\t\t  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");
21. printf("请选择:\n");
22. }
23. //////////////////
24. **void** cheng(**char** str1[],**char** str2[])//大数乘法函数
25. {
26. Bigint num1,num2,num3;
27. **int** i,j;
28. num1.len=strlen(str1);
29. memset(num1.digits,0,**sizeof**(num1.digits));
30. **for**(i=0;i<num1.len;i++)
31. num1.digits[i]=str1[num1.len-1-i]-'0';
32. num2.len=strlen(str2);
33. memset(num2.digits,0,**sizeof**(num2.digits));
34. **for**(i=0;i<num2.len;i++)
35. num2.digits[i]=str2[num2.len-1-i]-'0';
36. num3.len=num1.len+num2.len;
37. memset(num3.digits,0,**sizeof**(num3.digits));
38. **for**(i=0;i<num1.len;i++)
39. **for**(j=0;j<num2.len;j++)
40. num3.digits[i+j]=num1.digits[i]\*num2.digits[j];
41. **for**(i=0;i<num3.len;i++)
42. **if**(num3.digits[i]>9)
43. {
44. num3.digits[i+1]+=num3.digits[i]/10;
45. num3.digits[i]%=10;
46. **while**(num3.digits[num3.len])
47. num3.len++;
48. }
49. **for**(i=num3.len-1;i>=0;i--)
50. printf("%d",num3.digits[i]);
51. putchar('\n');
52. }
53. //////////////////
54. **void** jia(**char** str1[],**char** str2[])//大数加法函数
55. {
56. Bigint num1,num2,num3;
57. **int** i;
58. **int** maxlen;
59. num1.len=strlen(str1);
60. memset(num1.digits,0,**sizeof**(num1.digits));
61. **for**(i=0;i<num1.len;i++)
62. num1.digits[i]=str1[num1.len-1-i]-'0';
63. num2.len=strlen(str2);
64. memset(num2.digits,0,**sizeof**(num2.digits));
65. **for**(i=0;i<num2.len;i++)
66. num2.digits[i]=str2[num2.len-1-i]-'0';
67. **if**(num1.len>num2.len)
68. {
69. maxlen=num1.len;
70. num3.len=num1.len;
71. }
72. **else**
73. {
74. num3.len=num2.len;
75. maxlen=num2.len;
76. }
77. **for**(i=0;i<num3.len;i++)
78. num3.digits[i]=num1.digits[i]+num2.digits[i];
79. **for**(i=0;i<num3.len-1;i++)
80. **if**(num3.digits[i]>9)
81. {
82. num3.digits[i]=num3.digits[i]-10;
83. num3.digits[i+1]=num3.digits[i+1]+1;
84. }
85. **for**(i=num3.len-1;i>=0;i--)
86. printf("%d",num3.digits[i]);
87. putchar('\n');
88. }
89. //////////////////
90. **void** jian(**char** str1[],**char** str2[])//大数减法运算
91. {
92. Bigint num1,num2,num3;
93. **int** i;
94. **int** maxlen;
95. num1.len=strlen(str1);
96. memset(num1.digits,0,**sizeof**(num1.digits));
97. **for**(i=0;i<num1.len;i++)
98. num1.digits[i]=str1[num1.len-1-i]-'0';
99. num2.len=strlen(str2);
100. memset(num2.digits,0,**sizeof**(num2.digits));
101. **for**(i=0;i<num2.len;i++)
102. num2.digits[i]=str2[num2.len-1-i]-'0';
103. **if**(num1.len>num2.len)
104. {
105. maxlen=num1.len;
106. num3.len=num1.len;
107. }
108. **else**
109. {
110. num3.len=num2.len;
111. maxlen=num2.len;
112. }
113. **for**(i=0;i<num3.len;i++)
114. {
115. **if**(num1.digits[i]<num2.digits[i])
116. {
117. num1.digits[i]=num1.digits[i]+10;
118. num1.digits[i+1]=num1.digits[i+1]-1;
119. }
120. num3.digits[i]=num1.digits[i]-num2.digits[i];
121. }
122. **for**(i=0;i<num3.len-1;i++)
123. **if**(num3.digits[i]>9)
124. {
125. num3.digits[i]=num3.digits[i]-10;
126. num3.digits[i+1]=num3.digits[i+1]+1;
127. }
128. **for**(i=num3.len-1;i>=0;i--)
129. printf("%d",num3.digits[i]);
130. putchar('\n');
131. }
132. //////////////////
133. **int** cmp(**int** a[MAXLEN],**int** b[MAXLEN])//比较数的大小
134. {
135. **int** i,result;
136. result=0;
137. **for**(i=MAXLEN-1;i>=0;i--)
138. {
139. **if**(a[i]!=0&&b[i]==0)
140. **return** 1;
141. }
142. **for**(i=MAXLEN;i>=0;i--)
143. {
144. **if**(a[i]>b[i])
145. **return** 1;
146. **if**(a[i]<b[i])
147. **return** -1;
149. }
150. **return** 0;
151. }
152. //////////////////
153. **int** \* jian1(**char** str1[],**char** str2[])//大数减法运算（用于除法中的减法）
154. {
155. Bigint num1,num2,num3;
156. **int** i;
157. **int** \***INT**=NULL;
158. **int** maxlen;
159. **int** LEN;
160. memset(num3.digits,0,**sizeof**(num3.digits));
161. num1.len=strlen(str1);
162. memset(num1.digits,0,**sizeof**(num1.digits));
163. **for**(i=0;i<num1.len;i++)
164. num1.digits[i]=str1[num1.len-1-i]-'0';
166. num2.len=strlen(str2);
167. memset(num2.digits,0,**sizeof**(num2.digits));
168. **for**(i=0;i<num2.len;i++)
169. num2.digits[i]=str2[num2.len-1-i]-'0';
171. **if**(num1.len>num2.len)
172. {
173. maxlen=num1.len;
174. num3.len=num1.len;
175. LEN=num3.len;
176. }
177. **else**
178. {
179. num3.len=num2.len;
180. maxlen=num2.len;
181. LEN=num3.len;
182. }
183. **for**(i=0;i<num3.len;i++)
184. {
185. **if**(num1.digits[i]<num2.digits[i])
186. {
187. num1.digits[i]=num1.digits[i]+10;
188. num1.digits[i+1]=num1.digits[i+1]-1;
189. }
190. num3.digits[i]=num1.digits[i]-num2.digits[i];
191. }
192. num3.digits[i]=LEN;
193. **INT**=num3.digits;
194. **return** **INT**;
195. }
196. //////////////////
197. **void** division(**char** str1[],**char** str2[])//除法运算
198. {
199. Bigint num1,num2,num3;
200. **int** i,shang=0;
201. **int** \*ljian=NULL;
202. **int** lenlen;
203. **int** len1,len2;
204. len1=strlen(str1);
205. len2=strlen(str2);
206. **if**(len1>len2)
207. lenlen=len1;
208. **else**
209. lenlen=len2;
211. ljian=jian1(str1,str2);
212. lenlen=\*(ljian+lenlen);
213. \*(ljian+len2)=0;
214. ljian=NULL;
215. memset(num3.digits,0,**sizeof**(num3.digits));
217. num1.len=strlen(str1);
218. memset(num1.digits,0,**sizeof**(num1.digits));
219. **for**(i=0;i<lenlen;i++)
220. num1.digits[i]=str1[lenlen-1-i]-'0';
222. num2.len=strlen(str2);
223. memset(num2.digits,0,**sizeof**(num2.digits));
224. **for**(i=0;i<num2.len;i++)
225. num2.digits[i]=str2[num2.len-1-i]-'0';
226. ljian=jian1(str1,str2);
227. shang=shang+1;
228. **while**(1)
229. {
230. **for**(i=0;i<lenlen;i++)
231. num3.digits[i]=\*(ljian+i);
232. memset(str1,0,MAXLEN);
233. **for**(i=0;i<lenlen;i++)
234. str1[i]=num3.digits[lenlen-1-i]+48;
235. str1[i]='\0';
236. **if**(cmp(num3.digits,num2.digits)==1||cmp(num3.digits,num2.digits)==0)
237. shang=shang+1;
238. **else**
239. **if**(cmp(num3.digits,num2.digits)==-1)
240. **break**;
241. ljian=jian1(str1,str2);
242. }
243. printf("商  是：%d\n",shang);
244. printf("余数是：");
245. **for**(i=lenlen-1;i>=0;i--)
246. printf("%d",num3.digits[i]);
247. printf("\n");
248. }
249. //////////////////
250. **int** main()//main函数
251. {
252. **char** str1[MAXLEN];
253. **char** str2[MAXLEN];
254. **int** len1,len2;
255. **int** i;
256. format();
257. **char** zxl;//选择
258. **while**(1)
259. {
260. **switch**(zxl=getchar())
261. {
262. **case** '1'://加法
263. printf("请输入两个大数进行加法运算：\n");
264. scanf("%s",str1);
265. scanf("%s",str2);
266. printf("加法运算结果为：");
267. jia(str1,str2);
268. printf("\n");
269. memset(str1,0,MAXLEN);
270. memset(str2,0,MAXLEN);
271. **break**;
272. **case** '2'://减法
273. printf("请输入两个大数进行减法运算：\n");
274. scanf("%s",str1);
275. scanf("%s",str2);
276. len1=strlen(str1);
277. len2=strlen(str2);
278. printf("减法运算结果为：\n");
279. **if**(len1>len2)
280. jian(str1,str2);
281. **if**(len1<len2)
282. {
283. printf("-");
284. jian(str2,str1);
285. }
286. **if**(len1==len2)
287. **if** (str1[0]>=str2[0])
288. jian(str1,str2);
289. **else**
290. {
291. printf("-");
292. jian(str2,str1);
293. }
294. printf("\n");
295. **break**;
296. **case** '3'://乘法
297. printf("请输入两个大数进行乘法运算：\n");
298. scanf("%s",str1);
299. scanf("%s",str2);
300. printf("乘法运算结果为：");
301. cheng(str1,str2);
302. printf("\n");
303. memset(str1,0,strlen(str1));
304. memset(str2,0,strlen(str2));
305. len1=0;
306. len2=0;
307. **break**;
308. **case** '4'://除法
309. printf("请输入两个大数进行除法运算：\n");
310. scanf("%s",str1);
311. scanf("%s",str2);
312. len1=strlen(str1);
313. len2=strlen(str2);
314. printf("除法运算结果为：\n");
315. **if**(len1>len2)
316. division(str1,str2);
317. **if**(len1<len2)
318. {
319. printf("商  是：00\n");
320. printf("余数是：");
321. **for**(i=0;i<len2;i++)
322. printf("%c",str2[i]);
323. }
324. **if**(len1==len2)
325. **if** (str1[0]>=str2[0])
326. division(str1,str2);
327. **else**
328. {
329. printf("商  是：00\n");
330. printf("余数是：");
331. **for**(i=0;i<len2;i++)
332. printf("%c",str2[i]);
333. }
334. **break**;
335. **case** '5':
336. **goto** end;
337. **default**:
338. printf("输入错误!请重新输入!");
339. **break**;
340. }
341. printf("请选择:\n");
342. zxl=getchar();
343. }
344. end:    **return** 0;
345. }