```c
//Write an 8051 C program to send values 00 - FF to port P1.
//Solution:
#include <reg51.h>
void main(void)
{
unsigned char z;
for (z=0;z<=255;z++)
P1=z;
}

//Write an 8051 C program to send hex values for ASCII characters of
//0, 1, 2, 3, 4, 5, A, B, C, and D to port P1.
//Solution:
#include <reg51.h>
void main(void)
{
unsigned char mynum[]="012345ABCD";
unsigned char z;
for (z=0;z<=10;z++)
P1=mynum[z];
}

//Write an 8051 C program to toggle all the bits of P1 continuously.
//Solution:
//Toggle P1 forever
#include <reg51.h>
void main(void)
{
for (;;)
{
P1=0x55;
P1=0xAA;
}
}

//Write an 8051 C program to send values of -4 to +4 to port P1.
//Solution:
////Singed numbers
#include <reg51.h>
void main(void)
{
char mynum[]={+1,-1,+2,-2,+3,-3,+4,-4};
unsigned char z;
for (z=0;z<=8;z++)
P1=mynum[z];
}

//Write an 8051 C program to toggle bit D0 of the port P1 (P1.0)
//50,000 times.
//Solution:
#include <reg51.h>
sbit MYBIT=P1^0;
void main(void)
{
unsigned int z;
for (z=0;z<=50000;z++)
{
MYBIT=0;
MYBIT=1;
}
}

//Write an 8051 C program to toggle bits of P1 continuously forever
//with some delay.
//Solution:
////Toggle P1 forever with some delay in between
////"on" and "off"
#include <reg51.h>
void main(void)
{
unsigned int x;
for (;;) //repeat forever
```

```
73  {
74  P1=0x55;
75  for (x=0;x<40000;x++); //delay size
76  //unknown
77  P1=0xAA;
78  for (x=0;x<40000;x++);
79  }
80  }
81
82  //Write an 8051 C program to toggle bits of P1 ports continuously with
83  //a 250 ms.
84  //Solution:
85  #include <reg51.h>
86  void MSDelay(unsigned int);
87  void main(void)
88  {
89  while (1) //repeat forever
90  {
91  P1=0x55;
92  MSDelay(250);
93  P1=0xAA;
94  MSDelay(250);
95  }
96  }
97
98  void MSDelay(unsigned int itime)
99  {
100 unsigned int i,j;
101 for (i=0;i<itime;i++)
102 for (j=0;j<1275;j++);
103 }
104
105
106
107 //LEDs are connected to bits P1 and P2. Write an 8051 C program that
108 //shows the count from 0 to FFH (0000 0000 to 1111 1111 in binary)
109 //on the LEDs.
110 //Solution:
111 #include <reg51.h>
112 #define LED P2
113 void main(void)
114 {
115 P1=00; //clear P1
116 LED=0; //clear P2
117 for (;;) //repeat forever
118 {
119 P1++; //increment P1
120 LED++; //increment P2
121 }
122 }
123
124 //Write an 8051 C program to get a byte of data form P1, wait 1/2
125 //second, and then send it to P2.
126 //Solution:
127 #include <reg51.h>
128 void MSDelay(unsigned int);
129 void main(void)
130 {
131 unsigned char mybyte;
132 P1=0xFF; //make P1 input port
133 while (1)
134 {
135 mybyte=P1; //get a byte from P1
136 MSDelay(500);
137 P2=mybyte; //send it to P2
138 }
139 }
140
141 void MSDelay(unsigned int itime)
142 {
143 unsigned int i,j;
144 for (i=0;i<itime;i++)
```

```
145   for (j=0;j<1275;j++);
146   }
147
148
149   //Write an 8051 C program to get a byte of data form P0. If it is less
150   //than 100, send it to P1; otherwise, send it to P2.
151   //Solution:
152   #include <reg51.h>
153   void main(void)
154   {
155   unsigned char mybyte;
156   P0=0xFF; //make P0 input port
157   while (1)
158   {
159   mybyte=P0; //get a byte from P0
160   if (mybyte<100)
161   P1=mybyte; //send it to P1
162   else
163   P2=mybyte; //send it to P2
164   }
165   }
166
167   //Write an 8051 C program to toggle only bit P2.4 continuously without
168   //disturbing the rest of the bits of P2.
169   //Solution:
170   ////Toggling an individual bit
171   #include <reg51.h>
172   sbit mybit=P2^4;
173   void main(void)
174   {
175   while (1)
176   {
177   mybit=1; //turn on P2.4
178   mybit=0; //turn off P2.4
179   }
180   }
181
182
183   //Write an 8051 C program to monitor bit P1.5. If it is high, send 55H
184   //to P0; otherwise, send AAH to P2.
185   //Solution:
186   #include <reg51.h>
187   sbit mybit=P1^5;
188   void main(void)
189   {
190   mybit=1; //make mybit an input
191   while (1)
192   {
193   if (mybit==1)
194   P0=0x55;
195   else
196   P2=0xAA;
197   }
198   }
199
200
201   //A door sensor is connected to the P1.1 pin, and a buzzer is connected
202   //to P1.7. Write an 8051 C program to monitor the door sensor, and
203   //when it opens, sound the buzzer. You can sound the buzzer by
204   //sending a square wave of a few hundred Hz.
205   //Solution:
206   #include <reg51.h>
207   void MSDelay(unsigned int);
208   sbit Dsensor=P1^1;
209   sbit Buzzer=P1^7;
210   void main(void)
211   {
212   Dsensor=1; //make P1.1 an input
213   while (1)
214   {
215   while (Dsensor==1)//while it opens
216   {
```

```
217   Buzzer=0;
218   MSDelay(200);
219   Buzzer=1;
220   MSDelay(200);
221   }
222   }
223   }
224
225   void MSDelay(unsigned int itime)
226   {
227   unsigned int i,j;
228   for (i=0;i<itime;i++)
229   for (j=0;j<1275;j++);
230   }
231
232   //The data pins of an LCD are connected to P1. The information is
233   //latched into the LCD whenever its Enable pin goes from high to low.
234   //Write an 8051 C program to send “The Earth is but One Country” to
235   //this LCD.
236   //Solution:
237   #include <reg51.h>
238   #define LCDData P1 //LCDData declaration
239   sbit En=P2^0; //the enable pin
240   void main(void)
241   {
242   unsigned char message[] ="The Earth is but One Country";
243   unsigned char z;
244   for (z=0;z<28;z++) //send 28 characters
245   {
246   LCDData=message[z];
247   En=1; //a highEn=0; //-to-low pulse to latch data
248   }
249   }
250
251   //Write an 8051 C program to toggle all the bits of P0, P1, and P2
252   //continuously with a 250 ms delay. Use the sfr keyword to declare the
253   //port addresses.
254   //Solution:
255   //Accessing Ports as SFRs using sfr data type
256   sfr P0=0x80;
257   sfr P1=0x90;
258   sfr P2=0xA0;
259   void MSDelay(unsigned int);
260   void main(void)
261   {
262   while (1)
263   {
264   P0=0x55;
265   P1=0x55;
266   P2=0x55;
267   MSDelay(250);
268   P0=0xAA;
269   P1=0xAA;
270   P2=0xAA;
271   MSDelay(250);
272   }
273   }
274
275   void MSDelay(unsigned int itime)
276   {
277   unsigned int i,j;
278   for (i=0;i<itime;i++)
279   for (j=0;j<1275;j++);
280   }
281
282
283   //Write an 8051 C program to turn bit P1.5 on and off 50,000 times.
284   //Solution:
285   sbit MYBIT=0x95;
286   void main(void)
287   {
288   unsigned int z;
```

```
289   for (z=0;z<50000;z++)
290   {
291   MYBIT=1;
292   MYBIT=0;
293   }
294   }
295
296   //Write an 8051 C program to get the status of bit P1.0, save it, and
297   //send it to P2.7 continuously.
298   //Solution:
299   #include <reg51.h>
300   sbit inbit=P1^0;
301   sbit outbit=P2^7;
302   bit membit; //use bit to declare
303   //bit- addressable memory
304   void main(void)
305   {
306   while (1)
307   {
308   membit=inbit; //get a bit from P1.0
309   outbit=membit; //send it to P2.7
310   }
311   }
312
313   //Run the following program on your simulator and examine the results.
314   //Solution:
315   #include <reg51.h>
316   void main(void)
317   {
318   P0=0x35 & 0x0F; //ANDing
319   P1=0x04 | 0x68; //ORing
320   P2=0x54 ^ 0x78; //XORing
321   P0=~0x55; //inversing
322   P1=0x9A >> 3; //shifting right 3
323   P2=0x77 >> 4; //shifting right 4
324   P0=0x6 << 4; //shifting left 4
325   }
326
327   //Write an 8051 C program to toggle all the bits of P0 and P2
328   //continuously with a 250 ms delay. Using the inverting and Ex-OR
329   //operators, respectively.
330   //Solution:
331   #include <reg51.h>
332   void MSDelay(unsigned int);
333   void main(void)
334   {
335   P0=0x55;
336   P2=0x55;
337   while (1)
338   {
339   P0=~P0;
340   P2=P2^0xFF;
341   MSDelay(250);
342   }
343   }
344
345   void MSDelay(unsigned int itime)
346   {
347   unsigned int i,j;
348   for (i=0;i<itime;i++)
349   for (j=0;j<1275;j++);
350   }
351
352   //Write an 8051 C program to get bit P1.0 and send it to P2.7 after
353   //inverting it.
354   //Solution:
355   #include <reg51.h>
356   sbit inbit=P1^0;
357   sbit outbit=P2^7;
358   bit membit;
359   void main(void)
360   {
```

```
361   while (1)
362   {
363   membit=inbit; //get a bit from P1.0
364   outbit=~membit; //invert it and send it to P2.7
365   }
366   }
367
368
369   //Write an 8051 C program to read the P1.0 and P1.1 bits and issue an
370   //ASCII character to P0 according to the following table.
371   //P1.1 P1.0
372   //0 0 send '0' to P0
373   //0 1 send '1' to P0
374   //1 0 send '2' to P0
375   //1 1 send '3' to P0
376   //Solution:
377   #include <reg51.h>
378   void main(void)
379   {
380     unsignbed char z;
381     z=P1;
382     z=z&0x3;
383
384     switch (z)
385     {
386       case(0):
387       {
388       P0='0';
389       break;
390       }
391       case(1):
392       {
393       P0='1';
394       break;
395       }
396       case(2):
397       {
398       P0='2';
399       break;
400       }
401       case(3):
402       {
403       P0='3';
404       break;
405       }
406     }
407   }
408
409   //Write an 8051 C program to convert packed BCD 0x29 to ASCII and
410   //display the bytes on P1 and P2.
411   //Solution:
412   #include <reg51.h>
413   void main(void)
414   {
415   unsigned char x,y,z;
416   unsigned char mybyte=0x29;
417   x=mybyte&0x0F;
418   P1=x|0x30;
419   y=mybyte&0xF0;
420   y=y>>4;
421   P2=y|0x30;
422   }
423
424
425   //Write an 8051 C program to convert ASCII digits of '4' and '7' to
426   //packed BCD and display them on P1.
427   //Solution:
428   #include <reg51.h>
429   void main(void)
430   {
431     unsigned char bcdbyte;
432     unsigned char w='4';
```

```
433     unsigned char z='7';
434     w=w&0x0F;
435     w=w<<4;
436     z=z&0x0F;
437     bcdbyte=w|z;
438     P1=bcdbyte;
439   }
440
441   //Write an 8051 C program to calculate the checksum byte for the data
442   //25H, 62H, 3FH, and 52H.
443   //Solution:
444   #include <reg51.h>
445   void main(void)
446   {
447   unsigned char mydata[]={0x25,0x62,0x3F,0x52};
448   unsigned char sum=0;
449   unsigned char x;
450   unsigned char chksumbyte;
451   for (x=0;x<4;x++)
452   {
453   P2=mydata[x];
454   sum=sum+mydata[x];
455   P1=sum;
456   }
457   chksumbyte=~sum+1;
458   P1=chksumbyte;
459   }
460
461
462   //Write an 8051 C program to perform the checksum operation to
463   //ensure data integrity. If data is good, send ASCII character 'G' to P0.
464   //Otherwise send 'B' to P0.
465   //Solution:
466   #include <reg51.h>
467   void main(void)
468   {
469   unsigned char mydata[] = {0x25,0x62,0x3F,0x52,0xE8};
470   unsigned char chksum=0;
471   unsigned char x;
472   for (x=0;x<5;x++)
473   chksum=chksum+mydata[x];
474   if (chksum==0)
475   P0='G';
476   else
477   P0='B';
478   }
479
480
481   //Write an 8051 C program to convert 11111101 (FD hex) to decimal
482   //and display the digits on P0, P1 and P2.
483   //Solution:
484   #include <reg51.h>
485   void main(void)
486   {
487   unsigned char x,binbyte,d1,d2,d3;
488   binbyte=0xFD;
489   x=binbyte/10;
490   d1=binbyte%10;
491   d2=x%10;
492   d3=x/10;
493   P0=d1;
494   P1=d2;
495   P2=d3;
496   }
497
498
499   //Compile and single-step the following program on your 8051
500   //simulator. Examine the contents of the 128-byte RAM space to locate
501   //the ASCII values.
502   //Solution:
503   #include <reg51.h>
504   void main(void)
```

```
505   {
506   unsigned char mynum[]="ABCDEF"; //RAM space
507   unsigned char z;
508   for (z=0;z<=6;z++)
509   P1=mynum[z];
510   }
511
512
513   //Write, compile and single-step the following program on your 8051
514   //simulator. Examine the contents of the code space to locate the values.
515   //Solution:
516   #include <reg51.h>
517   void main(void)
518   {
519   unsigned char mydata[100]; //RAM space
520   unsigned char x,z=0;
521   for (x=0;x<100;x++)
522   {
523   z--;
524   mydata[x]=z;
525   P1=z;
526   }
527   }
528
529   //Compile and single-step the following program on your 8051
530   //simulator. Examine the contents of the code space to locate the ASCII
531   //values.
532   //Solution:
533   #include <reg51.h>
534   void main(void)
535   {
536   code unsigned char mynum[]="ABCDEF";
537   unsigned char z;
538   for (z=0;z<=6;z++)
539   P1=mynum[z];
540   }
541
542
543   //Write, compile and single-step the following program on your 8051
544   //simulator. Examine the contents of the code space to locate the values.
545   //Solution:
546   #include <reg51.h>
547   void main(void)
548   {
549   unsigned char mydata[100]; //RAM space
550   unsigned char x,z=0;
551   for (x=0;x<100;x++)
552   {
553   z--;
554   mydata[x]=z;
555   P1=z;
556   }
557   }
558
559
560   //Compile and single-step the following program on your 8051
561   //simulator. Examine the contents of the code space to locate the ASCII
562   //values.
563   //Solution:
564   #include <reg51.h>
565   void main(void)
566   {
567   code unsigned char mynum[]="ABCDEF";
568   unsigned char z;
569   for (z=0;z<=6;z++)
570   P1=mynum[z];
571   }
572
573
574   //Write a C program to send out the value 44H serially one bit at a time
575   //via P1.0. The LSB should go out first.
576   //Solution:
```

```c
#include <reg51.h>
sbit P1b0=P1^0;
sbit regALSB=ACC^0;
void main(void)
{
unsigned char conbyte=0x44;
unsigned char x;
ACC=conbyte;
for (x=0;x<8;x++)
{
P1b0=regALSB;
ACC=ACC>>1;
}
}


//Write a C program to send out the value 44H serially one bit at a time
//via P1.0. The MSB should go out first.
//Solution:
#include <reg51.h>
sbit P1b0=P1^0;
sbit regAMSB=ACC^7;
void main(void)
{
unsigned char conbyte=0x44;
unsigned char x;
ACC=conbyte;
for (x=0;x<8;x++)
{
P1b0=regAMSB;
ACC=ACC<<1;
}
}

//Write a C program to bring in a byte of data serially one bit at a time
//via P1.0. The LSB should come in first.
//Solution:
#include <reg51.h>
sbit P1b0=P1^0;
sbit ACCMSB=ACC^7;
bit membit;
void main(void)
{
unsigned char x;
for (x=0;x<8;x++)
{
membit=P1b0;
ACC=ACC>>1;
ACCMSB=membit;
}
P2=ACC;
}


//Write a C program to bring in a byte of data serially one bit at a time
//via P1.0. The MSB should come in first.
//Solution:
#include <reg51.h>
sbit P1b0=P1^0;
sbit regALSB=ACC^0;
bit membit;
void main(void)
{
unsigned char x;
for (x=0;x<8;x++)
{
membit=P1b0;
ACC=ACC<<1;
regALSB=membit;
}
P2=ACC;
}
```

```
649
650 //Example 9-20
651 //Write an 8051 C program to toggle all the bits of port P1 continuously
652 //with some delay in between. Use Timer 0, 16-bit mode to
653 //generate the delay.
654 //Solution:
655 #include <reg51.h>
656 void T0Delay(void);
657 void main(void){
658 while (1) {
659 P1=0x55;
660 T0Delay();
661 P1=0xAA;
662 T0Delay();
663 }
664 }
665 void T0Delay(){
666 TMOD=0x01;
667 TL0=0x00;
668 TH0=0x35;
669 TR0=1;
670 while (TF0==0);
671 TR0=0;
672 TF0=0;
673 }
674
675
676 //Example 9-21
677 //Write an 8051 C program to toggle only bit P1.5 continuously every
678 //50 ms. Use Timer 0, mode 1 (16-bit) to create the delay. Test the
679 //program on the (a) AT89C51 and (b) DS89C420.
680 //Solution:
681 #include <reg51.h>
682 void T0M1Delay(void);
683 sbit mybit=P1^5;
684 void main(void){
685 while (1) {
686 mybit=~mybit;
687 T0M1Delay();
688 }
689 }
690 void T0M1Delay(void){
691 TMOD=0x01;
692 TL0=0xFD;
693 TH0=0x4B;
694 TR0=1;
695 while (TF0==0);
696 TR0=0;
697 TF0=0;
698 }
699
700 //Example 9-22
701 //Write an 8051 C program to toggle all bits of P2 continuously every
702 //500 ms. Use Timer 1, mode 1 to create the delay.
703 //Solution:
704
705 #include <reg51.h>
706 void T1M1Delay(void);
707 void main(void){
708 unsigned char x;
709 P2=0x55;
710 while (1) {
711 P2=~P2;
712 for (x=0;x<20;x++)
713 T1M1Delay();
714 }
715 }
716 void T1M1Delay(void){
717 TMOD=0x10;
718 TL1=0xFE;
719 TH1=0xA5;
720 TR1=1;
```

```
721   while (TF1==0);
722   TR1=0;
723   TF1=0;
724   }
725
726   //Example 9-25
727   //A switch is connected to pin P1.2. Write an 8051 C program to
728   //monitor SW and create the following frequencies on pin P1.7:
729   //SW=0: 500Hz
730   //SW=1: 750Hz, use Timer 0, mode 1 for both of them.
731   //Solution:
732   #include <reg51.h>
733   sbit mybit=P1^5;
734   sbit SW=P1^7;
735   void T0M1Delay(unsigned char);
736   void main(void){
737   SW=1;
738   while (1) {
739   mybit=~mybit;
740   if (SW==0)
741   T0M1Delay(0);
742   else
743   T0M1Delay(1);
744   }
745   }
746
747   void T0M1Delay(unsigned char c){
748   TMOD=0x01;
749   if (c==0) {
750   TL0=0x67;
751   TH0=0xFC;
752   }
753   else {
754   TL0=0x9A;
755   TH0=0xFD;
756   }
757   TR0=1;
758   while (TF0==0);
759   TR0=0;
760   TF0=0;
761   }
762
763
764   //Example 9-23
765   //Write an 8051 C program to toggle only pin P1.5 continuously every
766   //250 ms. Use Timer 0, mode 2 (8-bit auto-reload) to create the
767   //delay.
768   //Solution:
769   #include <reg51.h>
770   void T0M2Delay(void);
771   sbit mybit=P1^5;
772   void main(void){
773   unsigned char x,y;
774   while (1) {
775   mybit=~mybit;
776   for (x=0;x<250;x++)
777   for (y=0;y<36;y++) //we put 36, not 40
778   T0M2Delay();
779   }
780   }
781   void T0M2Delay(void){
782   TMOD=0x02;
783   TH0=-23;
784   TR0=1;
785   while (TF0==0);
786   TR0=0;
787   TF0=0;
788   }
789
790
791   //Example 9-24
792   //Write an 8051 C program to create a frequency of 2500 Hz on pin
```

```
793   //P2.7. Use Timer 1, mode 2 to create delay.
794   //Solution:
795   #include <reg51.h>
796   void T1M2Delay(void);
797   sbit mybit=P2^7;
798   void main(void){
799   unsigned char x;
800   while (1) {
801   mybit=~mybit;
802   T1M2Delay();
803   }
804   }
805   void T1M2Delay(void){
806   TMOD=0x20;
807   TH1=-184;
808   TR1=1;
809   while (TF1==0);
810   TR1=0;
811   TF1=0;
812   }
813
814   //Example 9-26
815   //Assume that a 1-Hz external clock is being fed into pin T1 (P3.5).
816   //Write a C program for counter 1 in mode 2 (8-bit auto reload) to count
817   //up and display the state of the TL1 count on P1. Start the count at 0H.
818   //Solution:
819   #include <reg51.h>
820   void main(void){
821   T1=1;
822   TMOD=0x60;
823   TH1=0;
824   while (1) {
825   do {
826   TR1=1;
827   P1=TL1;
828   }
829   while (TF1==0);
830   TR1=0;
831   TF1=0;
832   }
833   }
834
835   //Example 9-27
836   //Assume that a 1-Hz external clock is being fed into pin T0 (P3.4).
837   //Write a C program for counter 0 in mode 1 (16-bit) to count the pulses
838   //and display the state of the TH0 and TL0 registers on P2 and P1,
839   //respectively.
840   //Solution:
841   #include <reg51.h>
842   void main(void){
843   T0=1;
844   TMOD=0x05;
845   TL0=0;
846   TH0=0;
847   while (1) {
848   do {
849   TR0=1;
850   P1=TL0;
851   P2=TH0;
852   }
853   while (TF0==0);
854   TR0=0;
855   TF0=0;
856   }
857   }
858
859
860   //Example 10-15
861   //Write a C program for 8051 to transfer the letter “A” serially at 4800
862   //baud continuously. Use 8-bit data and 1 stop bit.
863   //Solution:
864   #include <reg51.h>
```

```
865   void main(void){
866   TMOD=0x20; //use Timer 1, mode 2
867   TH1=0xFA; //4800 baud rate
868   SCON=0x50;
869   TR1=1;
870   while (1) {
871   SBUF='A'; //place value in buffer
872   while (TI==0);
873   TI=0;
874   }
875   }
876
877   //Example 10-16
878   //Write an 8051 C program to transfer the message “YES” serially at
879   //9600 baud, 8-bit data, 1 stop bit. Do this continuously.
880   //Solution:
881   #include <reg51.h>
882   void SerTx(unsigned char);
883   void main(void){
884   TMOD=0x20; //use Timer 1, mode 2
885   TH1=0xFD; //9600 baud rate
886   SCON=0x50;
887   TR1=1; //start timer
888   while (1) {
889   SerTx('Y');
890   SerTx('E');
891   SerTx('S');
892   }
893   }
894   void SerTx(unsigned char x){
895   SBUF=x; //place value in buffer
896   while (TI==0); //wait until transmitted
897   TI=0;
898   }
899
900   //Example 10-17
901   //Program the 8051 in C to receive bytes of data serially and put them
902   //in P1. Set the baud rate at 4800, 8-bit data, and 1 stop bit.
903   //Solution:
904   #include <reg51.h>
905   void main(void){
906   unsigned char mybyte;
907   TMOD=0x20; //use Timer 1, mode 2
908   TH1=0xFA; //4800 baud rate
909   SCON=0x50;
910   TR1=1; //start timer
911   while (1) { //repeat forever
912   while (RI==0); //wait to receive
913   mybyte=SBUF; //save value
914   P1=mybyte; //write value to port
915   RI=0;
916   }
917   }
918
919   //Example 10-19
920   //Write an 8051 C Program to send the two messages “Normal Speed”
921   //and “High Speed” to the serial port. Assuming that SW is connected
922   //to pin P2.0, monitor its status and set the baud rate as follows:
923   //SW = 0, 28,800 baud rate
924   //SW = 1, 56K baud rate
925   //Assume that XTAL = 11.0592 MHz for both cases.
926   //Solution:
927   #include <reg51.h>
928   sbit MYSW=P2^0; //input switch
929   void main(void){
930   unsigned char z;
931   unsigned char Mess1[]="Normal Speed";
932   unsigned char Mess2[]="High Speed";
933   TMOD=0x20; //use Timer 1, mode 2
934   TH1=0xFF; //28800 for normal
935   SCON=0x50;
936   TR1=1; //start timer
```

```
 937
 938   if(MYSW==0) {
 939   for (z=0;z<12;z++) {
 940   SBUF=Mess1[z]; //place value in buffer
 941   while(TI==0); //wait for transmit
 942   TI=0;
 943   }
 944   }
 945   else {
 946   PCON=PCON|0x80; //for high speed of 56K
 947   for (z=0;z<10;z++) {
 948   SBUF=Mess2[z]; //place value in buffer
 949   while(TI==0); //wait for transmit
 950   TI=0;
 951   }
 952   }
 953   }
 954
 955   //Example 10-20
 956   //Write a C program for the DS89C4x0 to transfer the letter "A" serially
 957   //at 4800 baud continuously. Use the second serial port with 8-bit data
 958   //and 1 stop bit. We can only use Timer 1 to set the baud rate.
 959   //Solution:
 960   #include <reg51.h>
 961   sfr SBUF1=0xC1;
 962   sfr SCON1=0xC0;
 963   sbit TI1=0xC1;
 964   void main(void){
 965   TMOD=0x20; //use Timer 1, mode 2
 966   TH1=0xFA; //4800 baud rate
 967   SCON=0x50; //use 2nd serial port SCON1
 968   TR1=1; //start timer
 969   while (1) {
 970   SBUF1='A'; //use 2nd serial port SBUF1
 971   while (TI1==0); //wait for transmit
 972   TI1=0;
 973   }
 974   }
 975
 976   //Example 10-21
 977   //Program the DS89C4x0 in C to receive bytes of data serially via the
 978   //second serial port and put them in P1. Set the baud rate at 9600, 8-bit
 979   //data and 1 stop bit. Use Timer 1 for baud rate generation.
 980   //Solution:
 981   #include <reg51.h>
 982   sfr SBUF1=0xC1;
 983   sfr SCON1=0xC0;
 984   sbit RI1=0xC0;
 985   void main(void){
 986   unsigned char mybyte;
 987   TMOD=0x20; //use Timer 1, mode 2
 988   TH1=0xFD; //9600 baud rate
 989   SCON1=0x50; //use 2nd serial port SCON1
 990   TR1=1; //start timer
 991   while (1) {
 992   while (RI1==0); //monitor RI1
 993   mybyte=SBUF1; //use SBUF1
 994   P2=mybyte; //place value on port
 995   RI1=0;
 996   }
 997   }
 998
 999   //Example 11-14
1000   //Write a C program that continuously gets a single bit of data from P1.7
1001   //and sends it to P1.0, while simultaneously creating a square wave of
1002   //200 µs period on pin P2.5. Use Timer 0 to create the square wave.
1003   //Assume that XTAL = 11.0592 MHz.
1004   //Solution:
1005   //We will use timer 0 mode 2 (auto-reload). One half of the period is
1006   //100 µs. 100/1.085 µs = 92, and TH0 = 256 - 92 = 164 or A4H
1007
1008   #include <reg51.h>
```

```
1009  sbit SW =P1^7;
1010  sbit IND =P1^0;
1011  sbit WAVE =P2^5;
1012  void timer0(void) interrupt 1 {
1013  WAVE=~WAVE; //toggle pin
1014  }
1015  void main() {
1016  SW=1; //make switch input
1017  TMOD=0x02;
1018  TH0=0xA4; //TH0=-92
1019  IE=0x82; //enable interrupt for timer 0
1020  while (1) {
1021  IND=SW; //send switch to LED
1022  }
1023  }
1024
1025
1026  //Example 11-16
1027  //Write a C program using interrupts to do the following:
1028  //(a) Receive data serially and send it to P0
1029  //(b) Read port P1, transmit data serially, and give a copy to P2
1030  //(c) Make timer 0 generate a square wave of 5 kHz frequency on P0.1
1031  //Assume that XTAL = 11.0592 MHz. Set the baud rate at 4800.
1032  //Solution:
1033  #include <reg51.h>
1034  sbit WAVE =P0^1;
1035  void timer0() interrupt 1
1036    {
1037      WAVE=~WAVE; //toggle pin
1038    }
1039
1040  void serial0() interrupt 4
1041    {
1042    if (TI==1)
1043      {
1044      TI=0; //clear interrupt
1045      }
1046    else
1047      {
1048      P0=SBUF; //put value on pins
1049      RI=0; //clear interrupt
1050      }
1051  }
1052
1053  void main()
1054  {
1055  unsigned char x;
1056  P1=0xFF; //make P1 an input
1057  TMOD=0x22;
1058  TH1=0xF6; //4800 baud rate
1059  SCON=0x50;
1060  TH0=0xA4; //5 kHz has T=200us
1061  IE=0x92; //enable interrupts
1062  TR1=1; //start timer 1
1063  TR0=1; //start timer 0
1064  while (1)
1065    {
1066    x=P1; //read value from pins
1067    SBUF=x; //put value in buffer
1068    P2=x; //write value to pins
1069    }
1070  }
1071
1072
1073  //Example 11-17
1074  //Write a C program using interrupts to do the following:
1075  //(a) Generate a 10 KHz frequency on P2.1 using T0 8-bit auto-reload
1076  //(b) Use timer 1 as an event counter to count up a 1-Hz pulse and
1077  //display it on P0. The pulse is connected to EX1.
1078  //Assume that XTAL = 11.0592 MHz. Set the baud rate at 9600.
1079  //Solution:
1080  #include <reg51.h>
```

```
1081  sbit WAVE =P2^1;
1082  unsigned char cnt;
1083
1084  void timer0() interrupt 1 {
1085  WAVE=~WAVE; //toggle pin
1086  }
1087
1088  void timer1() interrupt 3 {
1089  cnt++; //increment counter
1090  P0=cnt; //display value on pins
1091  }
1092
1093  void main() {
1094  cnt=0; //set counter to 0
1095  TMOD=0x42;
1096  TH0=0x46; //10 KHz
1097  IE=0x86; //enable interrupts
1098  TR0=1; //start timer 0
1099  while (1); //wait until interrupted
1100  }
1101
1102
1103  //Example 12-2
1104  //Write an 8051 C program to send letters 'M', 'D', and 'E' to the LCD
1105  //using the busy flag method.
1106  //Solution:
1107  #include <reg51.h>
1108  sfr ldata = 0x90; //P1=LCD data pins
1109  sbit rs = P2^0;
1110  sbit rw = P2^1;
1111  sbit en = P2^2;
1112  sbit busy = P1^7;
1113
1114  void MSDelay(unsigned int itime)
1115  {
1116  unsigned int i,j;
1117  for (i=0;i<itime;i++)
1118    for (j=0;j<1275;j++);
1119  }
1120
1121  void lcdready(){
1122  busy = 1; //make the busy pin at input
1123  rs = 0;
1124  rw = 1;
1125  while(busy==1){ //wait here for busy flag
1126  en = 0; //strobe the enable pin
1127  MSDelay(1);
1128  en = 1;
1129  }
1130  }
1131
1132  void lcdcmd(unsigned char value){
1133  lcdready(); //check the LCD busy flag
1134  ldata = value; //put the value on the pins
1135  rs = 0;
1136  rw = 0;
1137  en = 1; //strobe the enable pin
1138  MSDelay(1);
1139  en = 0;
1140  return;
1141  }
1142  void lcddata(unsigned char value){
1143  lcdready(); //check the LCD busy flag
1144  ldata = value; //put the value on the pins
1145  rs = 1;
1146  rw = 0;
1147  en = 1; //strobe the enable pin
1148  MSDelay(1);
1149  en = 0;
1150  return;
1151  }
1152
```

```
1153   void main(){
1154   lcdcmd(0x38);
1155   lcdcmd(0x0E);
1156   lcdcmd(0x01);
1157   lcdcmd(0x06);
1158   lcdcmd(0x86); //line 1, position 6
1159   lcddata('M');
1160   lcddata('D');
1161   lcddata('E');
1162   }
```