Microprocessor Codes:

1)CUBE OF A NUMBER

```
ORG 0000H
MOV A,20H
MOV B,A
MUL AB
MOV 21H,A
MOV 22H,B
MOV A,20H
MOV B,21H
MUL AB
MOV 23H,A
MOV 24H,B
MOV A,20H
MOV B,22H
MUL AB
MOV 25H,A
MOV 26H,B
MOV 30H,23H
MOV A,24H
MOV A,25H
MOV 31H,26H
ADDC A,#0H
MOV 32H,A
```

2)PUSH OR POP

```
ORG 0000H
```

```
MOV R0,#25H
MOV R1,#35H
MOV R2,#45H
MOV R3,#55H
MOV R4,#65H
PUSH 0
PUSH 1
PUSH 2
PUSH 3
PUSH 4
MOV R0,#00H
MOV R1,#00H
MOV R2,#00H
MOV R3,#00H
MOV R4,#00H
POP 4
POP 3
POP 2
POP 1
POP 0
END
```

3)FACTORIAL OF A NUMBER

```
ORG 0000H
MOV R1,#05H
MOV R7,#01H
LCALL FACT
MOV R7,A
FACT:
MOV A,R7
```

```
CJNE R1,#00,UP
SJMP UP1
UP:
MOV B,R1
MUL AB
DJNZ R1,UP
UP1:
RET
END
```

4)FIND THE LARGEST ELEMENT IN A ARRAY

```
ORG 0000H
MOV R0,#50H
MOV A,@R0
MOV R2,A
DEC R2
INC R0
MOV B,@R0
INC R0
BACK:MOV A,@R0
CJNE A,B,LOOP
JMP LOOP1
LOOP:JC LOOP 1
MOV B,A
LOOP1:INC R0
DJNZ R2,BACK
NEXT:MOV 60H,B
END
```

5)Copy string from ram 200h to ram 40h

```
ORG 0000H
MOV DPTR,#MYDATA
MOV R0,#40H
MOV R2,#13H
BACK: CLR A
MOVC A,@A+DPTR
MOV @R0,A
INC DPTR
INC R0
DJNZ R2,BACK
HERE :SJMP HERE
ORG 200H
MYDATA: DB “VITUNIVERSITY”
END
```

6)COPYING TO LOCATION 60H

```
ORG 0000H
MOV DPTR,#MYDATA
MOV R0,#40H
MOV R2,#11H
MOV R1,#60H
MOV R3,#11H
BACK:CLR A
MOVC A+@A+DPTR
MOV @R0,A
INC DPTR
INC R0
DJNZ R2,BACK
```

```
MOV R0,#40H
AGAIN: MOV A,@RO
MOV @R1,A
INC R0
INC R1
DJNZ R3,AGAIN
HERE: SJMP HERE
ORG 200H
MYDATA:DB “ANYA SOROHI”
END
```

7)EXCHANGE VALUES

```
ORG 0000H
MOV P1,#0FBH
MOV R0,#40H
MOV A,P1
LOOP:MOVB,#10
DIV AB
XCH A,B
ADD A,#30H
MOV @R0,A
XCH A,B
INC R0
JNZ LOOP
END
```

8)MOV VALUE 55H TO P1,P2,P3 SIMULTANEOUSLY

```
ORG 0000H
MOV A,#55H
AGAIN:MOV P0,A
MOV P1,A
MOV P2,A
MOV P3,A
ACALL DELAY
CPL A
SJMP AGAIN
DELAY:MOV R5,#2
HERE1:MOV R4,#180
HERE2:MOV R3,#255
HERE3:DJNZ R3,HERE3
DJNZ R4,HERE2
DJNZ R5,HERE1
RET
```

9)250 MICROSECS DELAY USING TIMERS

```
ORG 0000H
MOV TMOD,#10H
AGAIN: TL1,#1AH
MOV TH1, #0FFH
SETB TR1
BACK:JNB TF1,BACK
CLR TR1
CPL P1.5
CLR TF1
SJMP AGAIN
```

END

10)MOVING VALUES FROM P1 TO P2

```
ORG 0000H
MOV A,#0FFH
MOV P1,A
BACK: MOV A,P1
MOV P2,A
SJMP BACK
END
```

11)SERIAL COMMUNICATION- A

```
ORG 000H
SJMP MAIN
MAIN:MOV TMOD,#20H
MOV TH1,#-6
MOV SCON,#50H
SETB TR1
AGAIN: MOV SBUF,#”A”
HERE:JNB TI, HERE
CLR TI
SJMP AGAIN
END
```

12)VIT UNIVERSITY

```
ORG 0000H
SJMP MAIN
MAIN: MOV DPTR,#3000H
MOV TMOD,#20H
MOV RO,#0EH
MOV TH1,#0FDH
```

```
MOV SCON,#50H
SETB TR1
UP:CLR A
MOVC A,@A+DPTR
AGAIN:MOV SBUF,A
HERE:JNB T1, HERE
CLR TI
INC DPTR
DJNZ R0,UP
SJMP $
ORG 3000H
DB ‘VIT UNIVERSITY’
END
```

13)

```
ORG 0000H
MOV TMOD,#20H
MOV SCON, #50H
SETB TR1
HERE: JNB RI,HERE
MOV A,SBUF
MOV P1,A
CLR RI
SJMP HERE
END
```

14) PROGRAM THAT CONTINUOUSLY SENDS 8 BIT DATA FROM P0 TO P1 WHILE CREATING A SQUARE WAVE OF 200US PERIOD ON PIN P2.1. USE TIMER 0

```
ORG 0000H
LJMP MAIN
ORG 000BH
```

```
CPL P2.1
RETI
ORG 0030H
MAIN:MOV TMOD,#02H
MOV P0,#0FFH
MOV TH0,#-92
MOV IE,#82H
SETB TRO
BACK:MOV A, P0
MOV P1,A
SJMP BACK
END
```

15)GENERATE TWO WAVES FOR DUTY CYCLE 40% AND 70% USING INTERRUPT

```
ORG 0X000
MOV TMOD,#01H
MOV TH0,#0FDH
MOV TH0,#0F0H
MOV TH1,#0C9H
MOV TL1,#0C7H
MOV TCON,#52H
MOV P1,#00H
MOV P2,#00H
SETB ET0
SETB ET1
SETB EA
MAIN: LOOP:
SJMP MAIN-LOOP
TIME Q0-ISR
CLR TF0
MOV TH0,#0FDH
```

```
MOV TL0,#0FDH
CPL P1.0
RETI
TIME Q1-ISR
CLR TF1
MOV TH1,#0C7H
MOV TL1,#0C7H RETE
CPL P2.0
END
```

16)WAVEFORM THAT GENERATES (ON-50MS, OFF-80MS)

```
ORG 0000H
LJMP MAIN
ORG 000BH
DJNZ R0,TEN
MOV R0,#36H
JB P1.1,0HN
DJNZ R2,TEN
MOV R2,#8
SETB P1.1
CPL P0.1
SJMP TEN
CONN:DJNZ R1,TEN
MOV R1,#5
CPL P1.1
CPL P0.1
TEN RETI
ORG 0030H
MAIN:
MOV R0,#36
MOV R1,#5
MOV R2,#8
```

```
MOV TMOD,#24
MOV THO,#-255
MOV IE,#82H
SETB P1.1
SETB TR0
BACK:SJMP BACK
END
```