SAINTS TO SINNERS

Here is a musical test for the members of your family who have a keen ear for a tune.

The object of the game is to guess the tune being played and to make it easy, to begin with, we have allowed your computer to play the entire tune. After the first ten 'numbers' you will only hear a short snatch from the tune.

We have included a very large selection of tunes suitable for 'saints and sinners'.

To make life more difficult for the player we have entered the tunes using a special code, so that they cannot be guessed at in advance.

How to play

When you think that you have guessed the title of the tune being played correctly, type in the full title, press RETURN and find out if your ear is musical or tin.

Programming hints

Lines 200-480 contain the procedure that plays the tune. The notes of the tune are held in the first and second, if any, elements of the array A$ and the name of the tune is held in N$. W is the indicator determining which tune is to be played. The tune stored in array A$ is terminated by the letters XXXXX. If more than one element of the array is needed to store the tune, the first element is terminated by the letters NNNNN. Storing data in a string variable is a very useful trick when there are too many fields to be assigned to use DATA statements, as you will be accessing data randomly, not sequentially. See the section entitled Possible alterations for further details.

The obvious alterations that can be made are that when you know the names of the tunes you will want to change them. If you wish to increase the number of tunes that can be played then you must increase the maximum value of W held in line 100. You could then include your tunebetween 380 and 390 starting with a statement ensuring that the tune is skipped over if the value of W is ensuring that the tune is skipped over if the value of W is not the correct one. You could then work out the tune you require on an instrument, or else you could copy a musical score. If you are copying a musical score then you should refer to the User Guide, but if you are doing it for fun then I recommend a child's musical instrument which usually just has the octave which starts with middle

C which is the most common octave. The pitch numbers for this octave are:

Middle ....C 053
....D 061
....E 069
....F 073
....G 081
....A 089
....B 097
....C 101

The duration of the notes should be smaller numbers than specified in the User Guide as processing the array takes time. Therefore I suggest that the durations should be 03, 08 and 18 approximately for notes of short, middle or long duration. When you have worked out the pitch and duration of all the notes, you should assign them to the first and, if more room is required, the second element of the array. The pitch number must have three digits and the duration must have two and they should be joined together and separated from details of the next note by a space. As stated before, the first element is terminated by NNNNN and the second element by XXXXX.

I do not expect you will bother to put the name of the tune in code, but in case you do N$ is made up of the ASCII values of the letters of the name of the tune, remembering to include spaces which have an ASCII value of 32. If you do not bother to code the name of the tune, assign the name to NAM$ and make sure the program skips to line 120.

```
10   REM SAINTS TO SINNERS
20   REM COPYRIGHT (C) G.LUDINSKI 1983
30   MODE5:VDU23;8202;0;0;0;
40   DIM A$(2)
50   NT=0
```

```
   60  CLS
   70  NT=NT+1
   80  PRINT:PRINT" Saints to sinners":PR
OC_RADIO
   90  PROC_RADIO
  100  W=INT(RND(1)*5)
  110  PROC_TUNE(W)
  120  NAM$="":FOR I=1 TO LEN(N$) STEP 2:
NAM$=NAM$+CHR$(VAL(MID$(N$,I,2))):NEXT I
  130  PRINT:PRINT"What is this tune   ca
lled ?":PRINT
  140  INPUT I$
  150  IF I$=NAM$ THEN PRINT:PRINT"You ar
e right":GOTO180
  160  PRINT:PRINT"No,it is called ":PRIN
T NAM$
  170  PRINT:PRINT"Do you want more Y/N"
  180  INPUT I$:IF I$="Y" OR I$="" THEN G
OTO 70
  190  GOTO 600
  200  DEFPROC_TUNE(W)
  210  A$(1)="":A$(2)="":N$=""
  220  IF W<>0 THEN GOTO 250
  230  A$(1)="06105 08115 08915 09725 000
05 09705 10110 10105 09710 08105 09710 0
8910 00005 06105 06920 07720 08110 08905
 09705 06910 08915 10105 09720 08920 081
25 00005 XXXXX "
  240  N$="80766573837382326839657779858 2
"
  250  IF W<>1 THEN GOTO 290
  260  A$(1)=STRING$(2,"05310 06110 06910
 05310 00010 ")+STRING$(2,"06910 07310 0
8120 00005 ")+"NNNNN "
  270  A$(2)=STRING$(2,"08105 08905 08105
 07305 06910 05310 00010 ")+STRING$(2,"0
5310 04110 05320 00010 ")+"XXX"
  280  N$="70826982693274656781856983"
  290  IF W<>2 THEN GOTO 320
  300  A$(1)="06905 06105 05310 06105 069
05 07305 08110 00005 08905 09705 10110 0
9710 08910 08110 00010 08905 09705 10110
 09710 08910 08110 08910 09710 10110 081
10 07310 06910 00010 XXXXX "
  310  N$="84726932707382838432787969 76"
  320  IF W<>3 THEN GOTO 350
  330  A$(1)="05310 06910 07320 00005 073
05 00001 07305 06905 07305 08110 00001 0
8110 07320 00005 07310 08910 10110 00005
 08905 00001 08905 08105 07302 00001 073
02 06905 07320 XXXXX "
  340  N$="79786769327373832827989657632 68
658736839833267738489"
  350  IF W<>4 THEN GOTO 390
  360  A$(1)="06905 00001 06905 07310 081
20 00001 07305 06905 06110 00001 06110 0
6905 00001 06905 07310 08120 00010 06905
 00001 06905 07310 08120 00001 06905 073
10 08920 NNNNN "
  370  A$(2)="00010 08110 08910 10120 000
01 10105 09705 08910 00001 08910 00005 0
8105 07305 06910 00001 06910 XXX"
  380  N$="7469328439657377 69"
  390  IF NT>10 THEN A$(1)=RIGHT$(A$(1),6
*INT(RND(1)*17+3))
  400  FOR J=1 TO 2
  410  FOR I=1 TO 255 STEP 6
  420  IF MID$(A$(J),I,3)="NNN" THEN I=25
5:GOTO460
  430  IF MID$(A$(J),I,3)="XXX" THEN I=25
```

```
  5:J=2:GOTO460
   440  IF  MID$(A$(J),I,3)="000"  THEN  SOUN
D 1,0,0,VAL(MID$(A$(J),I+3,2)):GOTO460
   450  SOUND 1,-15,VAL(MID$(A$(J),I,3)),V
AL(MID$(A$(J),I+3,2))
   460  NEXT  I
   470  NEXT  J
   480  ENDPROC
   490  DEFPROC_RADIO
   500  GCOL0,1:PROCBLOCK(300,150,600,250)
   510  GCOL0,2:PROCBLOCK(330,330,540,50)
   520  GCOL0,3:PROCBLOCK(300,400,10,30)
   530  PROCBLOCK(300,430,600,10)
   540  PROCBLOCK(890,400,10,30)
   550  ENDPROC
   560  DEFPROCBLOCK(X,Y,W,H)
   570  MOVE  X,Y:MOVE  X+W,Y
   580  PLOT  85,X,Y+H
   590  PLOT  85,X+W,Y+H
   600  ENDPROC
   610  REM
```