Lire attentivement les deux programmes suivants permettant de lire et d'écrire une liste de vecteurs de longueur variable:

1. Décrivez en détail le code et son fonctionement.

2. Pour lire les vecteurs, il faut connaître à l'avance le nombre de vecteurs à lire. Comment faire pour lire tous les vecteurs présents, sans en connaître le nombre à l'avance ? (vous pouvez changer le format)

3. Comment faire pour ne lire qu'un seul vecteur (par exemple le troisième de la liste) ? Présentez une solution en pseudo-code qui ne nécéssite pas un changement de format. Indiquez les appels systèmes et fonctions employées.

readVec.c

```c
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <unistd.h>
#include <stdlib.h>
#include <stdio.h>

double *loadSingle( int fd, int *length ) {
  double *v;
  int l;
  read( fd, length, sizeof(int) );
  l = (*length) * sizeof(double);
  v = (double*) malloc(l);
  read( fd, v, l );
  return v;
}


int load( int n, double ***vecs, int **lengths, char *fname ) {
  int fd = open( fname, O_RDONLY );
  int iVec = 0;
  *vecs = calloc( n, sizeof(double*) );
  *lengths = calloc( n, sizeof(int) );
  for( iVec=0; iVec < n; iVec++ ) {
    (*vecs)[iVec] = loadSingle( fd, &( (*lengths)[iVec] )  );
  }
  close(fd);
  return 1;
}

int main( int argc, char **argv ) {
  double **vecs;
  int *lengths;
  int iVec;
  load( 3, &vecs, &lengths, "vec.dat" );
  for( iVec = 0; iVec < 3; iVec ++ ) {
    int i;
    for( i = 0; i < lengths[iVec]; i++ ) {
      printf("%f ", vecs[iVec][i] );
```

```
    }
    printf("\n");
  }
  exit(EXIT_SUCCESS);
}
```

writeVec.c

```
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <unistd.h>
#include <stdlib.h>
#include <stdio.h>


int storeSingle( double *vec, int length, int fd ) {
  write( fd, (char*) &length, sizeof(int) );
  write( fd, (char*) vec, length * sizeof(double) );
  return 1;
}

int store( double **vec, int *lengths, int n, char *fname ) {
  int fd = open( fname,O_WRONLY | O_CREAT | O_TRUNC, S_IRUSR | S_IWUSR )
      ;
  int iVec = 0;
  for( iVec=0; iVec < n; iVec++ ) {
    storeSingle( vec[iVec], lengths[iVec], fd );
  }
  close(fd);
  return 1;
}

int main( int argc, char **argv ) {
  int lengths[] = { 2, 8, 5 };
  double v1[] = {0.1, 0.5};
  double v2[] = {1.0, 2, 3, 4, 5, 6, 7, 8};
  double v3[] = {-100.0, -10, 1, 10, 100};
  double *vecs[3];
  vecs[0] = v1;
  vecs[1] = v2;
  vecs[2] = v3;
  store( vecs, lengths, 3, "vec.dat" );
  exit(EXIT_SUCCESS);
}
```

Makefile

```
CC=gcc
CFLAGS=-Wall -g --pedantic

all: writeVec.out readVec.out

writeVec.out: writeVec.o
	$(CC) $(CFLAGS) -o writeVec.out writeVec.o

```

```
writeVec.o: writeVec.c
	$(CC) $(CFLAGS) -c -o writeVec.o writeVec.c

readVec.out: readVec.o
	$(CC) $(CFLAGS) -o readVec.out readVec.o

readVec.o: readVec.c
	$(CC) $(CFLAGS) -c -o readVec.o readVec.c

clean:
	rm -rf *.o *.out *.dat
```