```c
#include <stdio.h>

#define MAX_LIST_SIZE _________________________________

typedef struct ListOfDoubles {
  int size;
  double contents[MAX_LIST_SIZE];
} ListOfDoubles;

ListOfDoubles concat(ListOfDoubles d1, ListOfDoubles d2) {
  ListOfDoubles result;
  result.size = d1.size + d2.size;
  for(int i = 0; i < d1.size; i += 1) {
    result.contents[i] = d1.contents[i];
  }
  for(int i = 0; i < d2.size; i += 1) {
    result.contents[i + d1.size] = d2.contents[i];
  }
  return result;
}

int main() {
  ListOfDoubles lod = { 2, { 3.0, 4.0 } };
  ListOfDoubles lod2 = { 3, { 6.0, 9.0, 10.0 } };
  ListOfDoubles added = concat(lod, lod2);
  for(int i = 0; i < added.size; i += 1) {
    printf("%f ", added.contents[i]);
  }
  printf("\n");
}
```

```
$ gcc -Wall list.c -o list
$ ./list
3.000000 4.000000 6.000000 9.000000 10.000000
```

```c
#include <stdio.h>
#include <stdlib.h>

typedef struct ListOfDoubles {
  int size;
  double* contents;
} ListOfDoubles;

ListOfDoubles concat(ListOfDoubles d1, ListOfDoubles d2) {
  ListOfDoubles result;
  result.size = d1.size + d2.size;

  result.contents = malloc(result.size * sizeof(double));

  for(int i = 0; i < d1.size; i += 1) {
    result.contents[i] = d1.contents[i];
  }
  for(int i = 0; i < d2.size; i += 1) {
    result.contents[i + d1.size] = d2.contents[i];
  }
  return result;
}

int main() {
  double init1[] = { 3.0, 4.0 };
  ListOfDoubles lod = { 2, init1 };
  double init2[] = { 6.0, 9.0, 10.0 };
  ListOfDoubles lod2 = { 3, init2 };
  ListOfDoubles added = concat(lod, lod2);
  for(int i = 0; i < added.size; i += 1) {
    printf("%f ", added.contents[i]);
  }
  printf("\n");
}
```

```
$ gcc -Wall list_heap.c -o list_heap
$ ./list_heap
3.000000 4.000000 6.000000 9.000000 10.000000
```