Princeton University
COS 217: Introduction to Programming Systems
The Usefulness of C Dynamic Memory Management

Sample program: Read a specified number of integers from stdin, and write them in reverse order to stdout.

Version 1

```
#include <stdio.h>
#define MAX_COUNT 100
int main(int argc, char *argv[])
{
   int piArray[MAX_COUNT];
   int iCount;
   int i;
   printf("How many ints do you want to enter?  ");
   scanf("%d", &iCount);
   for (i = 0; i < iCount; ++i)
      scanf("%d", &piArray[i]);
   printf("The ints in reverse order are:\n");
   for (i = iCount-1; i >= 0;  --i)
      printf("%d\n", piArray[i]);
   return 0;
}
```

Runtime error: If user enters more than 100 integers, memory will be corrupted.

Version 2

```
#include <stdio.h>

int main(int argc, char *argv[])
{
   int iCount;
   int i;
   printf("How many ints do you want to enter?  ");
   scanf("%d", &iCount);

   int piArray[iCount];

   for (i = 0; i < iCount; ++i)
      scanf("%d", &piArray[i]);
   printf("The ints in reverse order are:\n");
   for (i = iCount-1; i >= 0;  --i)
      printf("%d\n", piArray[i]);

   return 0;
}
```

Compiletime error in ISO C: Array length must be a constant expression.

(over)

Version 3

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

int main(int argc, char *argv[])

{
   int *piArray;
   int iCount;
   int i;

   printf("How many ints do you want to enter?  ");
   scanf("%d", &iCount);

   piArray = (int*)calloc(iCount, sizeof(*piArray));

   for (i = 0; i < iCount; ++i)
      scanf("%d", &piArray[i]);

   printf("The ints in reverse order are:\n");
   for (i = iCount-1; i >= 0;  --i)
      printf("%d\n", piArray[i]);

   free(piArray);
   return 0;
}
```

Works!