Dynamic Memory Allocation in C

```
#include <stdio.h>
#include <stdlib.h> // needed for malloc and free

1.  int main() {

2.      int* newptr;
3.      newptr = (int *) malloc( sizeof( int ) * 20 );
4.      newptr[3] = 10;
5.      printf("%d\n", *(newptr + 3));

6.      free(newptr);

7.  }
```

Line 2: A pointer to int is declared.

Line 3: The **malloc** function is used to allocate memory for 20 times the size of an int (80 bytes total). This reserves memory for the equivalent of a 20-element integer array.

malloc returns a void* which must be cast to a pointer of the desired type, in this case int*.

Line 4: The pointer may be used to access "array" elements with *pointer-subscript notation.*

Line 5: The pointer may also be used to access "array" elements using a dereferenced *pointer-offset notation.*

Line 6: The 80 bytes allocated to newptr are freed (released to the operating system for reuse) by using the **free** function.