

4. Storage allocation technique

(i) <u>Aim</u>:- Write a program in C to perform storage allocation using array.

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
  // This pointer will hold the
  // base address of the block created
  int* ptr;
  int n, i;
  // Get the number of elements for the array
  printf("Enter number of elements:");
  scanf("%d",&n);
  printf("Entered number of elements: %d\n", n);
  // Dynamically allocate memory using malloc()
  ptr = (int*)malloc(n * sizeof(int));
  // Check if the memory has been successfully
  // allocated by malloc or not
  if (ptr == NULL) {
     printf("Memory not allocated.\n");
     exit(0);
```

```
}
  else {
     // Memory has been successfully allocated
     printf("Memory successfully allocated using malloc.\n");
     // Get the elements of the array
     for (i = 0; i < n; ++i) {
       ptr[i] = i + 1;
     }
     // Print the elements of the array
     printf("The elements of the array are: ");
     for (i = 0; i < n; ++i) {
       printf("%d, ", ptr[i]);
     }
  }
  return 0;
}
```

OUTPUT

```
Enter number of elements: 5

Memory successfully allocated using malloc.

The elements of the array are: 1, 2, 3, 4, 5,
```