findTarget

Write a C program that reads and searches character strings. In the program, it contains the function findTarget() that searches whether a target name string has been stored in the array of strings. The function prototype is

```
int findTarget(char *target, char nameptr[][80], int size);
```

where *nameptr* is the array of strings, *size* is the number of names stored in the array and *target* is the target string. If the target string is found, the function will return its index location, or -1 if otherwise.
In addition, the program also contains the functions readNames() and printNames(). The function readNames() reads a number of names from the user. The function prototype is given as follows:

```
void readNames(char nameptr[][80], int *size);
```

where *nameptr* is the array of strings to store the input names, and *size* is a pointer parameter which passes the number of names to the caller. The function prototype of printNames() which prints the names is given as follows:

```
void printNames(char nameptr[][80], int size);
```

A sample program template is given below for testing the functions:

```
#include <stdio.h>
#include <string.h>
#define SIZE 10
#define INIT_VALUE 999
void printNames(char nameptr[][80], int size);
void readNames(char nameptr[][80], int *size);
int findTarget(char *target, char nameptr[][80], int size);
int main()
{
  char nameptr[SIZE][80], t[40], *p;
  int size, result = INIT_VALUE;
  int choice;

  printf("Select one of the following options: \n");
  printf("1: readNames()\n");
  printf("2: findTarget()\n");
  printf("3: printNames()\n");
  printf("4: exit()\n");
  do {
    printf("Enter your choice: \n");
    scanf("%d", &choice);
    switch (choice) {
      case 1:
```

```
            readNames(nameptr, &size);
            break;
          case 2:
            printf("Enter target name: \n");
            scanf("\n");
            fgets(t, 80, stdin);
            if (p=strchr(t,'\n')) *p = '\0';
            result = findTarget(t, nameptr, size);
            printf("findTarget(): %d\n",  result);
            break;
          case 3:
            printNames(nameptr, size);
            break;
        }
      } while (choice < 4);
      return 0;
    }
    void printNames(char nameptr[][80], int size)
    {
      /* Write your program code here */
    }
    void readNames(char nameptr[][80], int *size)
    {
      /* Write your program code here */
    }
    int findTarget(char *target, char nameptr[][80], int size)
    {
      /* Write your program code here */
    }
```

Some sample input and output sessions are given below:

(1) Test Case 1:

```
Select one of the following options:
1: readNames()()
2: findTarget()
3: printNames()
4: exit()
Enter your choice:
1
Enter size:
4
Enter 4 names:
Peter Paul John Mary
Enter your choice:
2
Enter target name:
```

John
findTarget(): 2
Enter your choice:
4

(2) Test Case 2:
Select one of the following options:
1: readNames()()
2: findTarget()
3: printNames()
4: exit()
Enter your choice:
1
Enter size:
5
Enter 5 names:
Peter Paul John Mary Vincent
Enter your choice:
2
Enter target name:
Jane
findTarget(): -1
Enter your choice:
4

(3) Test Case 3:
Select one of the following options:
1: readNames()()
2: findTarget()
3: printNames()
4: exit()
Enter your choice:
1
Enter size:
5
Enter 5 names:
Peter Paul John Mary Vincent
Enter your choice:
3
Peter Paul John Mary Vincent

(4) Test Case 4:
Select one of the following options:
1: readNames()()
2: findTarget()
3: printNames()
4: exit()
Enter your choice:

1
Enter size:
6
Enter 6 names:
Peter Paul John Mary Vincent Joe
Enter your choice:
2
Enter target name:
Joe
findTarget(): 5
Enter your choice:
4