

ENG EK 125 - Worksheet C Chapter 5A

Name: Ande Chen

Section: C1

1) For the following program, create a document that shows what the output would be, and also explains the output.

```
#include <stdio.h>

void changeit(char *);

int main()
{
    char letter,
        symbol = '!',
        *cptr;

    cptr = &letter;
    *cptr = 'w';
    printf("*cptr is %c\n", *cptr);
    printf("letter is %c\n", letter);
    printf("symbol is %c\n\n", symbol);

    cptr = &symbol;
    printf("*cptr is %c\n", *cptr);
    printf("letter is %c\n", letter);
    printf("symbol is %c\n\n", symbol);

    *cptr = 'x';
    printf("*cptr is %c\n", *cptr);
    printf("letter is %c\n", letter);
    printf("symbol is %c\n\n", symbol);

    cptr = (char *) malloc(sizeof(char));
    *cptr = '8';
    printf("*cptr is %c\n", *cptr);
    printf("letter is %c\n", letter);
    printf("symbol is %c\n", symbol);

    changeit(cptr);
    printf("*cptr is %c\n", *cptr);
    free cptr;
    return 0;
}

void changeit(char *fncptr)
{
    printf("The char is %c\n", *fncptr);
    fncptr = (char *) malloc(sizeof(char));
    *fncptr = '3';
}
```

```

1  #include <stdio.h>
2  #include <stdlib.h>
3
4  void changeit(char *);
5
6  int main()
7  {
8      char letter,           // Initialize a char variable letter
9          symbol = '!',      // Initialize a char variable symbol as '!'
10         *cptr;             // Initialize a char pointer variable cptr
11
12     cptr = &letter;         // cptr points to the address of letter
13     *cptr = 'w';           // cptr stores 'w' which is then passed to letter
14
15     printf("*cptr is %c\n", *cptr);
16     printf("letter is %c\n", letter);
17     printf("symbol is %c\n\n", symbol);
18
19     cptr = &symbol;         // cptr points to the address of symbol, cptr is now !
20     printf("*cptr is %c\n", *cptr);
21     printf("letter is %c\n", letter);
22     printf("symbol is %c\n\n", symbol);
23
24     *cptr = 'x';           // cptr gets 'x' which is passed to symbol
25     printf("*cptr is %c\n", *cptr);
26     printf("letter is %c\n", letter);
27     printf("symbol is %c\n\n", symbol);
28
29     cptr = (char *) malloc(sizeof(char)); // cptr points to a new allocated space in memory
30     *cptr = '8';           // cptr gets '8' which is passed to the new allocated empty space
31     printf("*cptr is %c\n", *cptr);
32     printf("letter is %c\n", letter);
33     printf("symbol is %c\n", symbol);
34
35     changeit(cptr);         // cptr is passed to function change it
36     printf("*cptr is %c\n", *cptr);
37     /* cptr is still '8' because the preallocated space that holds '3' only exists in the scope of the function */
38     free(cptr);            // Clear cptr
39     return 0;
40 }
41
42 void changeit(char *fncptr) // return nothing, accept a char pointer variable
43 {
44     printf("The char is %c\n", *fncptr); // *fncptr is cptr, which is still 8
45     fncptr = (char *) malloc(sizeof(char)); // cptr points to a new allocated space in memory
46     *fncptr = '3';           // *cptr gets '3', which is passed to new allocated space
47 }

```

OUTPUT:

```

*cptr is w
letter is w
symbol is !

*cptr is !
letter is w
symbol is !

*cptr is x
letter is w
symbol is x

*cptr is 8
letter is w
symbol is x
The char is 8
*cptr is 8

```

```

chena8@WIT45005
tro to Programm
$ ./WSC5A_1
*cptr is w
letter is w
symbol is !

*cptr is !
letter is w
symbol is !

*cptr is x
letter is w
symbol is x

*cptr is 8
letter is w
symbol is x
The char is 8
*cptr is 8

```

2) Add another function *changeittoo* to the above program, similar to the *changeit* function, but instead of passing *cptr*, pass *&cptr*.

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  void changeit(char *);
5  void changeittoo(char **);
6
7  int main()
8  {
9      char letter,
10         symbol = '!',
11         *cptr;
12
13     cptr = &letter;
14     *cptr = 'w';
15     printf("*cptr is %c\n", *cptr);
16     printf("letter is %c\n", letter);
17     printf("symbol is %c\n\n", symbol);
18
19     cptr = &symbol;
20     printf("*cptr is %c\n", *cptr);
21     printf("letter is %c\n", letter);
22     printf("symbol is %c\n\n", symbol);
23
24     *cptr = 'x';
25     printf("*cptr is %c\n", *cptr);
26     printf("letter is %c\n", letter);
27     printf("symbol is %c\n\n", symbol);
28
29     cptr = (char *) malloc(sizeof(char));
30     *cptr = '8';
31     printf("*cptr is %c\n", *cptr);
32     printf("letter is %c\n", letter);
33     printf("symbol is %c\n", symbol);
34
35     changeit(cptr);
36     printf("*cptr is %c\n", *cptr);
37     changeittoo(&cptr);
38     printf("***cptr is %c\n", *cptr);
39     free(cptr);
40
41     return 0;
42 }
43
44 void changeit(char *fncptr)
45 {
46     printf("The char is %c\n", *fncptr);
47     fncptr = (char *) malloc(sizeof(char));
48     *fncptr = '3';
49 }
50
51 void changeittoo(char **fncptr)
52 {
53     printf("The char is %c\n", **fncptr);
54     *fncptr = (char *) malloc(sizeof(char));
55     **fncptr = '3';
56 }
```

```
chena8@WIT45005
tro to Programm
$ ./WSC5A_2
*cptr is w
letter is w
symbol is !

*cptr is !
letter is w
symbol is !

*cptr is x
letter is w
symbol is x

*cptr is 8
letter is w
symbol is x
The char is 8
*cptr is 8
The char is 8
**cptr is 3
```

3) Write a program that declares two pointer variables, one that points to an integer location, and one that points to a character location. Call a function that will initialize both by dynamically allocating memory and then putting values in the allocated locations. From main, print these values. Note that you must use the call-by-reference method to initialize the pointer variables from the function.

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  void initptr(int **, char **);
5
6  int main()
7  {
8      int *iptr;
9      char *cptr;
10
11     initptr(&iptr, &cptr);
12     printf("iptr is %d.\n", *iptr);
13     printf("cptr is %c.\n", *cptr);
14
15     free(iptr);
16     free(cptr);
17
18     return 0;
19 }
20
21 void initptr(int **iptr, char **cptr)
22 {
23     *iptr = (int *) malloc(sizeof(int));
24     **iptr = 3;
25     *cptr = (char *) malloc(sizeof(char));
26     **cptr = 'x';
27 }
```

```
chena8@WIT45005
tro to Programmi
$ ./WSC5A_3
iptr is 3.
cptr is x.
```

4) Write a program that defines a type for a structure that stores information on a student in ENG EK 125. Declare two variables to be this structure type, and call a function to initialize both of the structure variables, using call-by-reference.

```
chena8@WIT450:~$  
tro to Program  
$ ./WSC5A_4  
U12345678 C1  
U87654321 C3
```