Name: Nguyen Duong

Code: 2044

Subject: C Programing

**Exercise 1-1**

1. pData = data01[6];

**Exercise 1-2**

1. The address of data02[10]
2. The address of data02[5]
3. 5
4. They are the same

**Exercise 1-3**

35

**Exercise 1-4**

&a, &b

**Exercise 2-1**

1. char c
2. data[i]
3. data[i]
4. /\* Change one char \*/
5. char StoC(char c) {
6. /\* Only alphabet \*/
7. if (c >= 'a' && c <= 'z') {
8. return (c - 32);
9. }
10. /\* Nothing change if it is uppercase\*/
11. if (c >= 'A' && c <= 'Z') {
12. return c;
13. }
14. /\* If it is not alphabet \*/
15. return 0;
16. }
17. /\* Changes small letter string to capital letter string, return 1 if changing successfull \*/
18. int changeString(char \*data) {
19. int i;
20. for (i = 0; data[i] != '\0'; i++) {
21. data[i] = StoC(data[i]);
22. if (data[i] == 0) {
23. return 0;
24. }
25. }
26. return 1;
27. }
28. void main() {
29. char data01[32];
30. scanf("%s", data01);
31. int isValid = changeString(&data01);
32. /\* If changing successfull \*/
33. if (isValid == 1) {
34. printf(data01);
35. printf("\n");
36. }
37. else {
38. printf("Only alphabet is accepted, try again!\n");
39. }
40. }

**Exercise 2-2**

1. int my\_strlen(char \*data) {
2. int i = 0;
3. for (i = 0; data[i] != '\0'; i++) {
4. /\* Do nothing \*/
5. }
6. return i;
7. }

**Exercise 2-3**

char \*my\_strstr(char \*str, char \*substr) {

int length = strlength(substr);

char \*ref = substr;

while (\*str && \*ref) {

if (\*str++ == \*ref) {

ref++;

}

if (!\*ref) {

return (str - length);

}

if (length == (ref - substr)) {

ref = substr;

}

}

return NULL;

}

**Exercise 3-1**