North South University

Department of Electrical and Computer Engineering

CSE 115L: Programming Language I Lab

Week 09 – Assignments

---

1. Write a program in C to store n elements in an array and print the elements using pointer.
2. Write a program in C to Calculate the length of the string using a pointer.
3. Write a program in C to swap elements using call by reference.
4. Write a program in C to find the factorial of a given number using pointers.
5. Write a program in C to count the number of vowels and consonants in a string using a pointer.
6. Write a program in C to print the elements of an array in reverse order.

What is the output of this C code ?

```c
#include <stdio.h>
void foo(int*);
int main()
{
    int i = 10;
    foo((&i)++);
}
void foo(int *p)
{
    printf("%d\n", *p);
}
```

a) 10
b) Some garbage value
c) Compile time error
d) Segmentation fault/code crash

What is the output of this C code ?

```c
#include <stdio.h>
void foo(float *);
int main()
{
    int i = 10, *p = &i;
    foo(&i);
}
void foo(float *p)
{
    printf("%f\n", *p);
}
```

a) 10.000000
b) 0.000000
c) Compile time error
d) Undefined behavior

What is the output of this C code ?

```
1.  #include <stdio.h>
2.  int main()
3.  {
4.      int i = 10;
5.      int *p = &i;
6.      foo(&p);
7.      printf("%d ", *p);
8.      printf("%d ", *p);
9.  }
10. void foo(int **const p)
11. {
12.     int j = 11;
13.     *p = &j;
14.     printf("%d ", **p);
15. }
```

a) 11 11 11
b) 11 11 Undefined-value
c) Compile time error
d) Segmentation fault/code-crash

Which of the following **are** correct syntaxes to send an array as a parameter to function :
a) func(&array);
b) func(array);
c) func(*array);
d) func(array[size]);

Determine output:

```
#include <stdio.h>
void main()
{
      char *p = NULL;
      char *q = 0;
      if(p)
            printf(" p ");
      else
            printf("nullp");
      if(q)
            printf("q");
      else
            printf(" nullq");
}
```

a) p q
b) Depends on the compiler
c) x nullq where x can be p or nullp depending on the value of NULL
d) nullp nullq

The declaration
int (*p) [5];
means :

a) p is one dimensional array of size 5, of pointers to integers.
b) p is a pointer to a 5 elements integer array.
c) The same as int *p[
d) None of these.

Comment on the following ?
const int *ptr;

a) We cannot change the value pointed by ptr.
b) We cannot change the pointer ptr itself.
c) Both of the above
d) We can change the pointer as well as the value pointed by it.

A function 'p' that accepts a pointer to a character as argument and returns a pointer to an array of integer can be declared as :

a) int(*p(char *))[]
b) int *p(char *)[]
c) int (*p) (char *)[]

None of these.