BITS PILANI, DUBAI CAMPUS
DUBAI INTERNATIONAL ACADEMIC CITY, DUBAI

FIRST SEMESTER 2023 – 2024
COURSE: CSF301 (Principles of Programming Languages)
COMPONENT: Tutorial Sheet 4b **DATE:** 17 October 2023

Q1. Write down errors, if any, that may be present in the following C codes snippet and give reasons.

a.

```
/* sample.c */
#include  <stdio.h>
void  main()
{
        int startno= 2, endno=100, step=2, i;
       int sum = zero;
for ( i = startno; i <= endno; i += step)
      sum = sum + i;
printf( "startno = %d\n", *startno);
printf( "endno  = %d\n", endno);
printf( "step = %d\n", step);
printf("sum = %s\n", sum);
}
```

Ans: int sum = zero;
printf("startno = %d\n", *startno);
printf("sum = %s\n", sum);

b.

```
#include <stdio.h>
int main()
{
  int sum=0; // variable initialization
  float k=1;
  for(int i=1;i<=10;i++); // logical error, as we put the semicolon after loop
  {
     sum=sum+k;
     k++;
  }
printf("The  value of sum is %d", sum);
   return 0;
}
```

Ans:
logical error, as we put the semicolon after loop

Q2. Write the meaning of the following pointer declarations:

a. int *p(int a);
b. int (*p) (int a) ;
c. int *(*p)(int (*a)[]);

d. int (*p[l0])(char a);

Ans:

int *p(int a);

indicates a function that accepts an integer argument, and returns a pointer to an integer.

int (*p) (int a) ;

indicates a pointer to a function that accepts an integer argument and returns an integer. In this last declaration, the first pair of parentheses is used for nesting, and the second pair is used to indicate a function.

int *(*p)(int (*a)[]);

In this declaration, (*p) (. . .) indicates a pointer to a function. Hence, int * (*p) (. . .) indicates a pointer to a function that returns a pointer to an integer. Within the last pair of parentheses (the function's argument specification), (*a) [ ] indicates a pointer to an array. Therefore, int (*a) [ ] represents a pointer to an array of integers. Putting the pieces together, (*p) (int (*a) [ ]) represents a pointer to a function whose argument is a pointer to an array of integers.

And finally, the entire declaration

int *(*p)(int (*a)[]);

represents a pointer to a function that accepts a pointer to an array of integers as an argument, and returns a pointer to an integer.

int (*p[l0])(char a); /* p is a 10-element array of pointers to functions;

each function accepts an argument which is a character, and

returns an integer quantity */

Q3. A C program contains the following statements.

```
-----------------------------------
int i, j = 25;
int *pi, *pj = &j;
.............................
*pj = j + 5;
i= *pj + 5;
pi = pj;
*pi = i+ j;
----------------------------------------
```

Suppose each integer quantity occupies 2 bytes of memory. If the value assigned to i begins at (hexadecimal) address F9C and the value assigned to j begins at address F9E, then

(a) What value is represented by &i?

(b) What value is represented by &j?

(c) What value is assigned to pj?

(d) What value is assigned to *pj?

(e) What value is assigned to i?

(f) What value is represented by pi?

(g) What final value is assigned to *pi?

(h) What value is represented by (pi + 2)?

(i) What value is represented by the expression (*pi + 2)?

Ans: a) F9C b) F9E c) F9E d) 30 e) 35 f) F9E g) 65 h) FA2 i) 67

Q4. What is output of following C code snippets,

a.

```
void main() {
        int a[] = {1,2,3,4,5}, *p;
        p = a;
        ++*p;
        printf("%d ", *p);
        p += 2;
        printf("%d ", *p);
        }
```

Ans: 2 3

b.

```
char *ptr;
char mystring[] = "abcdefg";
ptr = myString;
ptr += 5;
```

Ans: fg