Problem Solving with C

Quiz #2

Compiled by

M S Anand (anandms@pes.edu)

Text Book(s):

1. “How To Solve It By Computer”, R G Dromey, Pearson, 2011.
2. “The C Programming Language”, Brian Kernighan, Dennis Ritchie, 2nd Edition, Prentice Hall PTR, 1988.

Reference Book(s):

1. “Expert C Programming; Deep C secrets”, Peter van der Linden
2. “ The C puzzle Book”, Alan R Feuer

What is the output?

```
int main(void)
{
    int i = 0;
    while (i <= 4)
    {
       printf("%d", i);
       if (i > 3)
        goto inside_foo;
       i++;
    }
    getchar();
    return 0;
}

void foo()
{
   inside_foo:
     printf("PP");
}
```

Output:
Compiler error: Label “inside_foo” used but not defined.
Explanation: Scope of a label is within a function. We cannot goto a label from other function.

What is the output of this program?

```
int main(void)
{
     char str[] = "pesuniversity";
     char *s1 = str, *s2 = str;
     int i;
    for(i = 0; i < 7; i++)
     {
        printf(" %c ", *str);
        ++s1;
     }
    for(i = 0; i < 6; i++)
     {
        printf(" %c ", *s2);
        ++s2;
      }
      getchar();
      return 0;
}
```

Output

```
p p p p p p p p e s u n i
```

Explanation

Both s1 and s2 are initialized to str. In first loop str is being printed and s1 is being incremented, so first loop will print only p. In second loop s2 is incremented and s2 is printed so second loop will print “p e s u n i ”

What is the output of this program?

```
int main(void)
{
    char str[] = “pesuforall";
    int i;
    for(i=0; str[i]; i++)
        printf("\n%c%c%c%c", str[i], *(str+i), *(i+str), i[str]);

    getchar();
    return 0;
}
```

Output:

```
pppp
eeee
ssss
uuuu
ffff
oooo
rrrr
aaaa
llll
```

Explanation:

Following are different ways of indexing both arrays and strings.

```
arr[i]
*(arr + i)
*(i + arr)
i[arr]
```

So all of them print same character.

What is the output of this program?

```
#include <stdio.h>

int main(void)
{
  char arr[]  = "abcdefghijk";
  char *ptr  = arr;

  while(*ptr != '\0')
      ++*ptr++;
  printf("%s %s", arr, ptr);

  getchar();
  return 0;
}
```

Answer: bcdefghijkl

Explanation:

Related to precedence and associativity
Below is the precedence of operators.

Postfix ++	left-to-right
Prefix ++	right-to-left
Dereference *	right-to-left

Therefore the expression ++*ptr++ has the following effect
Value of *ptr is incremented
Value of ptr is incremented

What is the output of this program?

```
#include <stdio.h>
int main(void)
{
  char arr[] = “abcdefghijklm";
  printf("%d", sizeof(arr));
  getchar();
  return 0;
}
```

Output: 14
The string “abcdefghijklm” has 13 characters, but the size is 14 because compiler includes a single ‘\0’ (string terminator) when char array size is not explicitly mentioned.

```
int main(void)
{
  int x, y = 5, z = 5;
  x = y==z;
  printf("%d", x);

  getchar();
  return 0;
}
```

The crux of the question lies in the statement x = y==z. The operator == is executed before = because precedence of comparison operators (<=, >= and ==) is higher than assignment operator =. The result of a comparison operator is either 0 or 1 based on the comparison result. Since y is equal to z, value of the expression y == z becomes 1 and the value is assigned to x via the assignment operator.

```
#include<stdio.h>

int main()
{
  int a[] = {1, 2, 3, 4, 5, 6};
  int *ptr = (int*)(&a+1);
  printf("%d ", *(ptr-1) );
  getchar();
  return 0;
}
```

Output: 6

&a is address of the whole array a[]. If we add 1 to &a, we get “base address of a[] + sizeof(a)”. And this value is typecast to int *. So ptr – 1 points to last element of a[]

What will be the output of the following C code?

```
#include <stdio.h>
int main(void)
{
    enum {ORANGE = 5, MANGO, BANANA = 4, PEACH};
    printf("PEACH = %d\n", PEACH);
    return 0;
}
```

a) PEACH = 3

b) PEACH = 4

c) PEACH = 5

d) PEACH = 6

What will be the output of the following C code?

```
#include <stdio.h>
int main(void)
{
    int a[5] = {1, 2, 3, 4, 5};
    int i;
    for (i = 0; i < 5; i++)
        if ((char)a[i] == '5')
            printf("%d\n", a[i]);
        else
            printf("FAIL\n");
    return 0;
}
```

Output:
FAIL 5 times

What will be the output of the following C code?

```
#include <stdio.h>
   int main(void)
   {
      float f1 = 0.1;
      if (f1 == 0.1)
         printf("equal\n");
      else
         printf("not equal\n");
      return 0;
   }
```

a) equal

b) not equal

c) output depends on compiler

d) none of the mentioned

What will be the output of the following C code?

```
#include <stdio.h>
int main(void)
{
    float x = 'a';
    printf("%f", x);
    return 0;
}
```

a) a

b) run time error

c) a.0000000

d) 97.000000

How many times will PESU be printed?

```
#include <stdio.h>
int main(void)
{
    int x;

    for (x=-1; x<10;x++)
    {
        if (x<5)
            continue;
        else
            break;
        printf("PESU\n");
    }
    return 0;
}
```

0

What is the output of the following program?

```
#include <stdio.h>
int main(void)
{
    int x=20,y=35;
    x=y++ + x++;
    y=++y + ++x;
    printf("%d  %d",x,y);
    return 0;
}
```

56 93

How will you print “Hello World” without semicolon?
Ans:

```
#include <stdio.h>
int main(void)
{
    if (printf("Hello World"))
    {
    }
    return 0;
}
```