Roll: _____　　　**Name:** _____

$\left[\right.$ *Write your answers in question paper. Answer <u>all</u> questions. All questions use C programming language.* $\left.\right]$

Q1. If the following C programs terminate, then write the output. Otherwise, say that the program does not terminate and cannot provide any output.　　　　　　　　　　　　　　　　　　　**(1.5×6)**

```c
#include <stdio.h>
int main () {
  int x = 20, y, z;
  z = ++x + (y = 10);
  printf ("x = %d ", x);
  printf ("y = %d ", y);
  printf ("z = %d ", z);
  return 0;
}
```

Answer:　　x = 21 y = 10 z = 31

```c
#include <stdio.h>
int main () {
  int x = 0;
  if ( x = 4 )  printf ("1 ");
  else if ( x > 0 )  printf ("2 ");
  if  ( !x )  printf ("3 ");
  if ( x - 5 )  printf ("4 ");
  return 0;
}
```

Answer:　　1 4

```c
#include <stdio.h>
int main () {
  int i = 0;
  while ( ++i < 5 )
    printf ("%d ", i++);
  return 0;
}
```

Answer:　　1 3

```c
#include <stdio.h>
int main () {
  for ( int i = 0; i < 5;
        i += (i%2 == 0)? 2 : 1 )
    printf("%d ", i);
  return 0;
}
```

Answer:　　0 2 4

```c
#include <stdio.h>
int main () {
  int i, count = 0;
  for ( i = 1; i <= 11; i++ ) {
    if ( i%2 == 0 ) i = i+1;
    else i = i-4;
    count++;
  }
  printf("count = %d", count);
  return 0;
}
```

Answer:　　count = 8

```c
#include <stdio.h>
int main () {
  int i = 11;
  while ( i == 11 ) {
    if ( i%2 == 0 ) i = i+1;
    else i = i-1;
    i++;
  }
  printf("i = %d\n", i);
  return 0;
}
```

Answer:　　No termination / output

Q2. What will be displayed if the following statements are executed from a C program? **(1.5)**

```
float num = -123.4567;
printf ("num = %2.2f", num);
```

Answer: num = -123.46

Q3. What will be displayed if the following print statement is executed from a C program? **(1.5)**

```
printf ("\"-5\'s\"\\n");
```

Select the correct answer.

(A) \   **(B)**   "-5's"\n   **(C)** It will give syntax error!   **(D)** \"-5\'s\"\ and go into newline

Q4. Which of the following can be valid name(s) of a C variable? *Select the correct answer(s).* **(1)**

(A) `default-2024`   **(B)** `2024default`   **(C)** `default`   **(D)** `_default`

Q5. How many different *odd* binary numbers can be represented in 1 Byte? (Recall that, 1 Byte = 8 Bits) **(1)**

Answer: $2^8/2$ = 2^7 = 128

Q6. A hiker climbs 100 meters uphill each day but descends 50 meters at night. Consider the C program given below which determines the number of days it will take for the hiker to reach a mountain peak that is 500 meters above the starting point. The code is also printing the hiker's elevation at the end of every day. Some sections of the code are left blank. Fill in these blanks appropriately to ensure that the program produces the desired result. **(2)**

```
#include <stdio.h>
int main ()
{
  int climbPerDay = 100, descendPerNight = 50;
  int targetElevation = 500, currentElevation = 0;
  int days = 0;

  do {
      currentElevation += climbPerDay;

      if ( currentElevation >= targetElevation )

                          break;

      currentElevation -=              descendPerNight               ;

      ++days;

      printf ("Day %d:  Hiker\'s elevation = %d meters.\n", days, currentElevation);

  } while (           currentElevation < targetElevation           );

  printf ("Hiker took %d days to reach mountain peak.\n",     days + 1     );

  return 0;
}
```

Q7. Consider a memory space of 96 Mega Bits. Assume that, a variable of `float` data type consumes 4 bytes of memory space and that of a `double` data type takes 8 bytes. Suppose, you wish to declare equal number of `float` as well as `double` variables through your program. What is the maximum number of variables that could be declared using the given memory space? *Show your calculations.*

Hint: 1 Mega Bit $= 2^{20}$ Bits and 1 Byte = 8 Bits **(2)**

Answer:

Let, N = number of `float` variables = number of `double` variables

Therefore, we have: $8 \times 8 \times N + 4 \times 8 \times N = 96 \times 2^{20} \implies N = 2^{20}$

So, the maximum number of variables that can be declared $= 2 \times N = 2^{21}$.

Q8. Consider that a *circle C* is represented by the coordinates of its *centre* and its *radius*. Suppose we have two circles: C_1 has its centre at (`x1,y1`) and has radius of `r1`. Circle C_2 has its centre at (`x2,y2`) and has radius of `r2`. A point (x, y) is *accepted* if it lies inside of only one of the circles, but not inside both circles. *Complete the following C program* to check whether a given point (x, y) is accepted. **(2)**

```c
#include <stdio.h>
#include <math.h>
int main ()
{
  /* declaration of variables */
  double x1, y1, r1;     // centre and radius of first circle C₁
  double x2, y2, r2;     // centre and radius of second circle C₂
  double x, y;           // point coordinates
  double d1, d2;         // distance of the point from centres of two circles

  /* taking input:  centre and radius of both circles and the point */
  scanf("%lf%lf%lf", &x1, &y1, &r1);
  scanf("%lf%lf%lf", &x2, &y2, &r2);
  scanf("%lf%lf", &x, &y);

  /* calculate distance of the given point from the centre of first circle */

  d1 =       sqrt( (x - x1) * (x - x1) + (y - y1) * (y - y1) )       ;

  /* calculate distance of the given point from the centre of second circle */

  d2 =       sqrt( (x - x2) * (x - x2) + (y - y2) * (y - y2) )       ;

  /* check whether the point lies inside only one circle, but not both */

  if ( (   (d1 < r1) && (d2 >= r2)   ) || (   (d1 >= r1) && (d2 < r2)   ) )

      printf ("Accepted\n");

  return 0;
}
```

— The question paper ends here. —

ROUGH WORK