For all questions, assume the size of the integer type and the address is 32-bits.

8 marks

Question 1: Copy your answer for each of the following questions to the table:

1	2	3	4	5	6	7	8	9	10
B ✓	A ✓	B ✓	C ✓	A ✗	B ✓	C ✓	B ✓	B ✓	D ✗

1. ..... is a storage class that references an identifier/variable that is defined somewhere else.

A. `auto` **B.** `extern` **C.** `static` **D.** `register`

---

2. The..... of an identifier is the program segment in which the identifier can be referenced.

A. scope **B.** storage class **C.** module **D.** None of the given

---

3. What is the output of the following program?

```
#include <stdio.h>
void f (){
  static int x=0;
  x+=10;
  printf("%d,",x);
}
int main(){
  f(); f(); return 0;
}
```

A. `0,10,` **B.** `10,20,` **C.** `10,10,` **D.** `20,20,`

---

4. What is the output of the following code segment?

```
char value='b';
char *colorStr = value=='R'?"Red":value=='B'?"Blue":"Unknown";
printf("%s",colorStr);
```

A. `Red` **B.** `Blue` **C.** `Unknown` **D.** None of the given

---

5. What is the output of the following code segment?

```
char c='A';
{ c++;
  char c='b';}
printf("%c", c);
```

A. `A` **B.** `B` **C.** `a` **D.** `b`

---

6. What is the output of the following code segment?

```
int i = 0;
for (; i ; )
printf("out1");
printf("out2");
```

A. `out1` **B.** `out2` **C.** `out1out2` **D.** None of the given

---

7. Casting a variable from its declared type to another data type ..... cause a precision loss.

A. will always **B.** will never **C.** may **D.** None of the given

---

8. What is the final value of x when the following code is run?

```
int x;
for (x=0; x<20; ++x) { }
```

A. `19` **B.** `20` **C.** `21` **D.** None of the given

9. What is the output of the following program (if any)?

```
#include <stdio.h>
int main(){
  const int N = 3;
  int i, j;
  for (i = 1; i <= N + 1; i++)
    for (j = 1; j <= N; j++)
      printf("%c%s", 'A' + ((i + j - 2) % N), j==N?"\n":"");
  return 0;
}
```

A.
```
ABC
BCA
CAB
```

B.
```
ABC
BCA
CAB
ABC
```

C.
```
ABC
BCD
CDE
```

D.
```
ABC
BCD
CDE
DEF
```

10. What is the output of the following program (if any)?

```
#include <stdio.h>
int var = 20;
int main(){
  int var = var/2;
  printf("%d", var);
  return 0;
}
```

A. 20 **B.** 10 **C.** undefined value **D.** compilation error

5 / 5 marks

Question 2: Write a statement to achieve each of the following tasks:

1. Define a symbolic constant `SIZE` that has a value 5 using const keyword.

   const int size = 5; ✓

2. Define an array named *numbers* with `SIZE` elements of type float.

   float numbers[SIZE]; ✓

3. Assign the value `3.46` to the second element in the array in section B.

   numbers[1] = 3.46; ✓

4. Print the second array element with `1` digit of precision to the right of the decimal point.

   printf("%.1f", numbers[1]); ✓

5. Declare a String named *str* and initialize it to literal value: **Summer**

   char str[] = "summer"; ✓

6. Write the function prototype for a function called `Mid` that takes a String as a parameter and returns a pointer to the middle character in the String.

   char* Mid(char*str); ✓

3 / 5 marks

Question 3: For each statement, show the output into the corresponding boxes.

1. `printf("%-4d%5d", 2, 4);`

2								4				

2. `printf("%+10.3e", -627.14);`

6	.	2	7	e	+	0	2					

3. `printf("%7.2f", 0.888);`

			0	.	8	9						

4. `printf("%-5.2f %.2f", 5.0, 123.4);`

5	.	0	0		1	2	3	.	4	0		

5. `printf("%f%c%d", 23.12, '+', 15);`

2	3	.	1	2	+	1	5					

4 / 5 marks

Question 4: Complete the following recursive function that determines whether or not a string `sub` is a subsequence of search string `str`. For instance: `"abcd"` is a subsequence of `"aebfcsd"`, `""` (empty string) is a subsequence of any string, `"abcde"` is a subsequence of `"abcde"`, `"abcd"` is a subsequence of `"aabbccdd"`, `"abc"` is NOT a subsequence of `"bca"` and .

```
int isSub(char* sub, char* str, int subIdx, int strIdx){

  /* base case for when we exhaust the sub string */
  if( sub[subIdx]=='\0' )  return 1 ;

  /* base case for when we exhaust the search string */
  if( str[strIdx]=='\0' )  return 0 ;

  /* recursive call:
     possible match between substring and search string */
  if( sub[subIdx]==str[strIdx] )
    return isSub( char* sub, char* str, subIdx++, strIdx++ );
  else /* recursive call: No match! move on */
    return isSub( char* sub, char* str, subIdx, strIdx++ );

}
```