TUTORIAL 7 (SECJ1013)

PROGRAMMING TECHNIQUE 1

SECTION 04 & 07, SEM 1, 2024/2025

Q#1

Given an incomplete program below, complete the program by filling in the blanks with an appropriate C++ statement according to the instructions stated in the comments.

```
1   #include <iostream>
2   using namespace std;
3
4   /*a) Declare prototype void function named getinput which has a
5   double pointer as a parameter*/
6   _____________________________________________________
7
8   /*b) Declare prototype void function named calcdiscount which has a
9   double pointer and float pointer to constant as parameters*/
10  _____________________________________________________
11
12  int main()
13  {
14     double a;
15     const float discount = 0.2;
16
17     getinput(&a);
18     calcdiscount(&a, &discount);
19     cout << "/nThe total price is: RM" << a;
20
21     return 0;
22  }
23
24  /*c) Define the getinput function to get the price from the user
25  input. Please use a parameter named b*/
26  ____________________ c(i) ____________________________ {
27    cout << "Please insert a price: RM";
28      ____________________ c(ii) ____________________
29  }
30
31  /*d) Define the calcdiscount function to calculate a price after
32  discount. Please use a double pointer parameter named b, and float
33  pointer parameter named d. If the price is more than RM 100, then
34  the rate is following the discount rate defined in main function. If
35  the price less than or equal to RM 100, then the rate is 10%
36  discount*/
37  ____________________ d(i) ____________________________ {
38    if (____________________ d(ii) ____________________________)
39        ____________________ d(iii) ____________________________
40    else
41        ____________________ d(iv) ____________________________
42  }
43
```

Q#2

Given the declaration for a structured data type named **S_TYPE** as follows.

```
1   struct S_TYPE
2   {
3      float fixedBefore;
4      CHOICE choice;
5      U_TYPE flexible;
6      float fixedAfter;
7   };
```

Declare further:

a) the data type **CHOICE** which is an enumerated data type whose members are **ONE** and **TWO**.

b) the data type **U_TYPE** which is a union data type which can be made up of either **choice1** a character data type or **choice2** an integer data type.

Q#3

A structure type called **SalesRecord** is designed to hold all necessary data regarding a salesman in a company and the sales information that he or she has made. The **SalesRecord** structure and an array variable called **salesRep** are declared below:

```
1   struct SalesRecord
2   {
3      int id;
4      char name[30];
5      double quarterlySales[4];
6      double totalAnnual;
7   };
8
9   SalesRecord salesRep[20];
```

a) A function named **readRecord** is designed to read the information of all the salesmenof the company from the user. Assume that the company has 20 salesmen. Below is the prototype of the function.

```
void readRecord (SalesRecord []);
```

i) Write a complete definition for the function **readRecord** to read the salesman's id, name and quarterly sales.

ii) Write a proper call statement to the function **readRecord** to fill in the array **salesRep.**

b) The member **totalAnnual** of **SalesRecord** is used to hold the information about the total amount of yearly sales of each salesman respectively. The calculation of the total sale is accomplished by the following function:

```
1   double getAnnualSale (double quarterlySales[])
2   {
```

```
3     double total = 0.0;
4     for (int i=0; i< 4; i++)
5     total += quarterlySales[i];
6     return total;
7  }
```

Write a call statement to this function to calculate the annual sale for every salesman in the list.

c) A function named **`getSalesman`** is used to search for the record of a salesman using the salesman's id from the salesmen list of record. The following program segment is written to call to this function. Define the function **`getSalesman`** in order to accomplish its purpose.

```
1  :
2  SalesRecord searchRecord:
3  int id;
4  :
5  cout << "Enter Salesman's ID =>" << endl; cin >> id;
6  searchRecord = getSalesman(salesRep, id);
7  :
8  :
```

Q#4

Given the following program segment, show the values of **`vals`** and **`temp`** variables, as well as values stored in **`val`** array after the program is executed. Write your answer in **Table 4**.

```
1  int val[] = {14, 9, 5, 2};
2  int *vals = val+1, *temp;
3
4  temp = vals + 2;
5  *temp = *--vals;
6  val[2] = *(vals+1) + 7;
7  *vals += 3;
8  *(val+1) = *vals + 5;
```

refer to first array

Table 4

	[0]	[1]	[2]	[3]	vals	temp
val	14	9	5	2	d4	dc
Address	*0x70fdd0*	*0x70fdd4*	*0x70fdd8*	*0x70fddc*	*0x70fdc8*	*0x70fdc0*

Q#5

Based on the program segment below, what is the output for the program if the following statement is placed in the blank space in **Line 6**?

```
1  int a = 7;
2  int b = 8;
```

```
3   int *ptr1 = &a;
4   int *ptr2 = &b;
5
6   if (__________________)
7     cout << "Value in a is bigger than b";
8   else
9     cout << "Value in b is bigger than a";
```

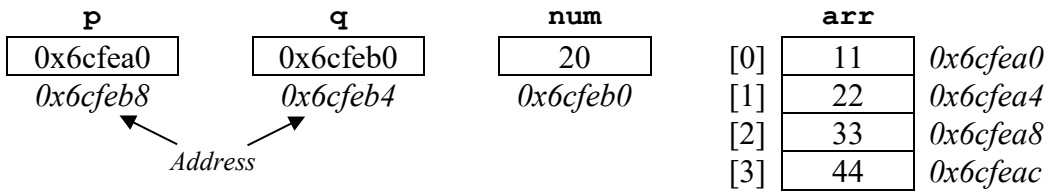
a) **`*ptr1 > *ptr2`**
Answer: _____

b) **`ptr2 = ptr1`**
Answer: _____

Q#6

a) Based on the **Program 4** and memory layout given in **Figure 2** below that utilizes pointers, write the output of the program.

```
1   //Program 4
2   #include <iostream>
3   using namespace std;
4
5   int main()
6   {
7       int *p, *q;
8       int num = 20;
9       int arr[] = {11, 22, 33, 44};
10
11      p = arr;
12      cout << p + 1 << endl;
13      cout << &p << endl;
14      cout << *(p + 3) << endl;
15      cout << arr << endl;
16      cout << ++(*p) << endl;
17
18      q = &num;
19      cout << *q << endl;
20      cout << &q << endl;
21      cout << q << endl;
22
23      return 0;
24  }
```



***Note**:* The values in *italic* are the address of the memory

Figure 2: Memory layout

b) Based on the comments given (**bold** text) in the **Program 5**, complete the program with appropriate C++ statements. ***Note:*** You may need to write more than one C++ statements to answer some of the questions given in the comments.

```
1   //Program 5
2   #include <iostream>
3   using namespace std;
4   const int MAXNAME = 10;
5
6   int main()
7   {
8      int pos;
9      char *name = NULL;
10     int *one = NULL;
11     int *two = NULL;
12     int result;
13
14     //(i) Dynamically allocate memory to the integer pointer
15     //variables named one and two
16     ____________________________________________________________
17
18     //(ii) Dynamically allocate an array with size MAXNAME
19     //to the character pointer variable named name
20     ____________________________________________________________
21
22     cout << "Enter your last name with exactly 10 characters."
23          << endl;
24
25     for (pos = 0; pos < MAXNAME; pos++)
26        //(iii) Read a character using pointer from the keyboard
27        //and insert it into the name array
28        ____________________________________________________________
29
30     cout << endl << "Enter two integer numbers" << endl;
31
32     //(iv) Read two integer numbers using pointers
33     //from the user
34     ____________________________________________________________
35
36     //(v) Calculate the sum of the two numbers using pointers
37     //and store it in the variable named result
38     ____________________________________________________________
39
40     cout << "The sum of the three values is " << result
41          << endl;
42
43     //(vi) Free all dynamically allocated memory
44     ____________________________________________________________
45
46     return 0;
47  }
```

Q#7

Write the output of the code segment below.

```
float number = 150.50, number2=85.50;
```

```
    float * pNumber;
    pNumber = &number;  // address of number is 0x2c

    cout << pNumber  << endl;
    cout << &number  << endl;
    cout << *pNumber << endl;
    cout << number   << endl;

    *pNumber = *pNumber + 90.50;
    cout << ++*pNumber << endl;
    cout << --number   << endl;

    pNumber = &number2;
    number=*pNumber + 20.20;
    cout << *pNumber  << endl;
    cout << number    << endl;
```

Q#8

a. Given the following program (**Program 3**) in **Table 2**. Show the values of **`ptr`** and **`temp`** variables, as well as values stored in **`Arr`** array in the blank boxes in **Figure 2**. The first **`Arr`** array in **Figure 2** is when the program is executed until Line 11, while the second **`Arr`** array in **Figure 2** is when the program is executed until Line 18.

Table 2

Program 3		**Output**
1	`#include <iostream>`	
2	`using namespace std;`	
3		
4	`int main()`	
5	`{`	
6	`   int Arr[] = {25, 30, 45, 70};`	
7	`   int *ptr = &Arr[0], *temp;`	
8		
9	`   temp = ptr + 2;`	
10	`   *temp = *++ptr;`	
11	`   *ptr = *Arr+2;`	
12	`   Arr[3] = *(ptr+1) + 5;`	
13	`   *Arr += 3;`	
14		
15	`   cout << _____ << endl;`	**0x72fe20**
16	`   cout << _____ << endl;`	**0x72fe34**
17	`   cout << _____ << endl;`	**30**
18	`   return 0;`	
19	`}`	

Answer:

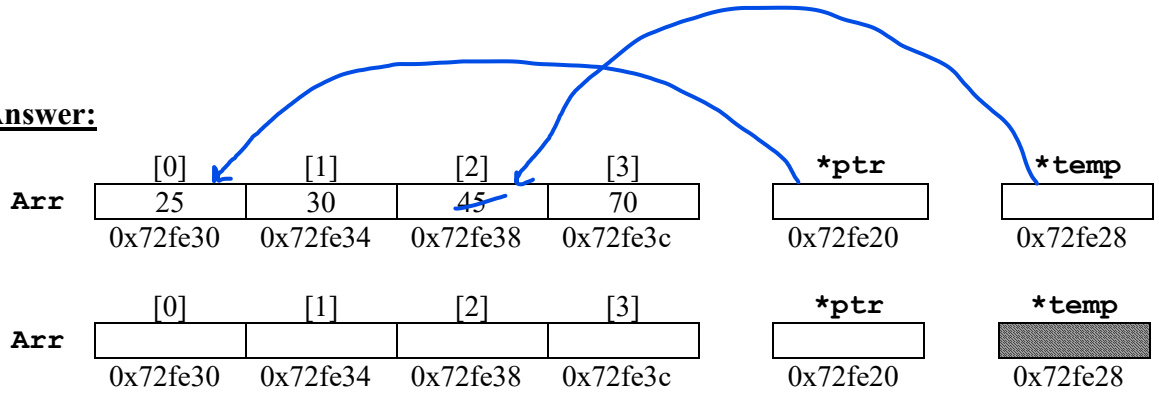


Figure 2

b. Based on output given of **Program 3** in **Table 2**, complete the C++ statements in line 15 to 17. ***Note:*** You should only use **`ptr`** variable to answer this question. Write your answers in **Table 3**.

Answer:

Table 3

Line	C++ statements
15	`cout << _____ << endl;`
16	`cout << _____ << endl;`
17	`cout << _____ << endl;`