

1. Find the output of the following programs

[2+4]

(a) #include <stdio.h>

```
int main() {
int int_array[3] = {4,9,3};
int *ptr1,*ptr2,*ptrs[3];
ptr1 = &int_array[2];
ptr2 = &int_array[0];
ptrs[0]=ptr1;
ptrs[1]=ptr2;
ptrs[2]=&int_array[1];
printf("The values are %d %d %d\n",*ptrs[0],*ptrs[1],*ptrs[2]);
}
```

(b) #include <stdio.h>

```
int main() {
char *ptr;
char arrayChars[10] = {'E','x','c','e','l','l','e','n','t','\0'};
ptr = arrayChars;
printf("The word reads %s.\n", arrayChars);
*ptr = 'e';
printf("Now it reads %s.\n", arrayChars);
printf("The 4th character of the array is %c.\n",*(ptr +=3));
*(ptr+1) = ' ';
*(ptr+2) = ' ';
printf("Now the word reads %s.\n", arrayChars);
}
```

2. Write a C program which declares an array of 10 floating point numbers and initialises the array to the multiples of 8: 8.0, 16.0, 24.0, 32.0, ...

[3]

3. Write a program that reads a number that says how many integer numbers are to be stored in an array, creates an array to fit the exact size of the data and then reads in that many numbers into the array. [5]

4. Write a C function called **add6** which takes an integer parameter and adds 6 to it. **add6** should change the value of the parameter in both the called and calling functions. [5]

5. `int x=4;`

`int* intPtr;`

`int y;`

`intPtr=&x;`

`printf(“%d\n”,*intPtr);`

(a) What value does the above code print out?

(b) Take the value in x, add 3 to it, and assign that to y. Use intPtr in your expression. [1+2]

6. Write statements to do the following

a) Declare a 26-element character array named **f**.

b) Declare a character **ch**.

c) Read a character into the first element of **f**.

d) Copy that character into every other element of **f**.

e) Change the 7th element of **f** to the character ‘x’. [1+1+1+1+1]

7. The following recursive function operates on an integer array of length n, returning an integer. What does it do? [3]

```
int mystery(int a[], int n){
if (n==1) return a[0]*a[0];
else
return a[n-1]*a[n-1]+mystery(a,n-1);
}
```

8. For each of the following, write a single C statement that performs the indicated task.

- a) Declare and initialise **letter** to be a character variable with the value 'B'.
- b) Declare and initialise **ch** to be a character variable with the value 'y'.
- c) Declare **ptr** to be a pointer to objects of type char.
- d) Assign the address of **ch** to the variable **ptr**.
- e) Assign the address of **letter** to the variable **ptr**.
- f) Change the value of the object pointed to by **ptr** to 'd'. [1+1+1+1+1+1]

9. Write statement to do the following

- (a) Declare a struct called **courseRec** with four components: (i) a string containing the subject code, exactly 6 characters, eg. MTH409, (ii) a string containing the subject title, max 30 characters, eg. Computer programming in C, (iii) a int containing the subject credit, eg, 4, and (iv) an int containing the no of students enrolled, eg. 47
- (b) Declare a variable **subj** of the struct type declared in (a).
- (c) Read the subject code of **subj** from the keyboard.
- (d) Read the student enrollment and assign 4 to the credit of **subj**.
- (e) Read the subject title of **subj** from the keyboard.
- (f) Print out the components of **subj**, as shown in the example below

MTH409 Computer programming in C 4 47 [3+1+1+2+2+2]

10. Write a C function which takes an integer parameter (greater than 1), and returns the highest power of that integer which is less than 1,000. [eg. for parameter value 2 your function should return 512 (since $2^9 = 512 < 1000$ but $2^{10} > 1000$). Similarly for parameter values 3 and 1234, your function should return 729 and 1] [3]