The University of Nottingham Ningbo China

SCHOOL OF COMPUTER SCIENCE

A LEVEL 1 MODULE, AUTUMN SEMESTER 2020-2021

PROGRAMMING AND ALGORITHMS (COMP1038)

Time allowed: 1.0 hour (60 minutes)

Candidates may complete the front cover of their answer book and sign their desk card but must NOT write anything else until the start of the examination period is announced.

Answer ALL questions.

No calculators are permitted in this examination.

Dictionaries are not allowed with one exception. Those whose first language is not English may use a standard translation dictionary to translate between that language and English provided that neither language is the subject of this examination. Subject specific translation dictionaries are not permitted.

No electronic devices capable of storing and retrieving text, including electronic dictionaries, may be used.

DO NOT turn examination paper over until instructed to do so.

ADDITIONAL MATERIAL: None.

INFORMATION FOR INVIGILATORS: Collect both the exam papers and the answer booklets at the end of exam.

Question 1: What is the difference between call by value and call by reference in C programming? (2 marks)

Question 2: What does dynamic memory allocation mean? What are functions in C programming to implement dynamic memory allocation? Use C code to show how to allocate and release memory dynamically for an integer array that contains 1000 elements. (2 marks)

Question 3: Explain what would happen when the program exam.c, as shown in Figure 1, is compiled using the command as shown in Figure 2 on cslinux. (2 marks)

Open

exam.c
~/Desktop/exam_2020

```
1 #include <stdio.h>
2
3 int main(void)
4 {
5         double a = 5.0;
6
7         printf("%lf\n", sqrt(a));
8
9         return 0;
10 }
```

Figure 1: exam.c

```
[z2017233@CSLinux Desktop]$ cd exam_2020
[z2017233@CSLinux exam_2020]$ pwd
/home/nottingham.edu.cn/z2017233/Desktop/exam_2020
[z2017233@CSLinux exam_2020]$ ls
exam.c
[z2017233@CSLinux exam_2020]$ gcc exam.c -o exam
```

Figure 2: Compiling on cslinux

Question 4: Use while loop to implement the same output as the following code. (2 marks)

```
for(a=1; a<=100; a++)
        printf("%d\n", a*a);
```

Question 5: Point out mistakes contained in the below program and suggest how to correct it. (4 marks).

```
#include <stdio.h>

struct emp
{
        char n[20];
        int age;
};

int main(void)
{
        struct emp e1 = {"David", 23};
        struct emp e2;
        scanf("%s %d", e2.n, e2.age);
        if(structcmp(e1, e2) == 0)
                printf("Equal structs.");
        else
                printf("Unequal structs.");
}

structcmp (struct emp x, struct emp y)
{
        if(x.n == y.n)
                if(x.age == y.age)
                        return 0;
        return 1;
}
```

Question 6: Complete the function body of *check* in the program below. This function should take two arguments, an integer called *input* and an integer pointer called *ptr*. A calculation to determine whether the *input* is odd, even or zero must be done within this function. If *input* is zero, the program should print 0. If *input* is odd, the program prints -1. If *input* is even, the

program prints 1. You can assume that the user will only enter a single whole number at each prompt (ie, no text, no floating-point numbers, no empty input, etc) (3 marks)

```
#include <stdio.h>

void check(int input, int *ptr);

int main(void)
{
	int in = 0;
	int out = 0;

	while(in != -9999)
	{
		scanf("%d", &in);
		check(in, &out);
		printf("%d\n", out);
	}
	return 0;
}
```

Question 7: An incomplete program on singly linked list is given below. Complete the printList function which prints out all values in the linked list. Each value should be printed in a line. (3 marks)

```
struct node{
	int data;
	struct node *next;
};
typedef struct node Node;

Node *head;

void InsertNodeAtFront(int data);
void printList();

void InsertNodeAtFront(int data)
{
   Node *temp = head;
   temp->data = data;
   temp->next = head;
   head = temp;
}
```

Question 8: Point out potential mistakes contained in the function 'func' and suggest how to correct them. (3 marks)

```
#include <stdio.h>
#include <stdlib.h>
struct ex
{
        int i;
        float j;
        char *s;
};

void func (void)
{
   struct ex *p = malloc(sizeof(struct ex));
   p->s = malloc(20 * sizeof(char));
   scanf("%s", &p->s);
   printf("%s", p->s)
   free(p);
}
```

Question 9: What is the output of the following program? (4 marks)

```
#include<stdio.h>
#include<stdbool.h>

int f1( int x, int y)
{
          x=x+2;
          y=y+3;
          return x+y;
}

int f2(  int *x, int y)
{
          *x=*x+2;
          y=y+3;
          return *x+y;
}

int f3 ( int *x, int *y)
{
```

```
        *x=*x+2;
        *y=*y+3;
        return *x+*y;
}

int f4( int x,  int *y, int *z)
{
        x=x+*y;
        *y=*z+3;
        z=&x;
        *z=*y*2;
        return *z;
}

int main(int argc, char *argv[])
{
        int k=1, m=3, r=2;
        printf("1) %d %d %d \n", k, m, r);
        r=f1(k, m);
        printf("2) %d %d %d \n", k, m, r);
        r=f2(&k, m);
        printf("3) %d %d %d \n", k, m, r);
        r=f3(&k, &m);
        printf("4) %d %d %d \n", k, m, r);
        r=f4(k, &m, &r);
        printf("5) %d %d %d \n", k, m, r);
        return 0;
}
```