

## Указания за контролното за заверка

- Време за работа – 90 мин.
- Всяка задача се оценява с 1 точка за правилен отговор и 0 точки за грешен
- Точките от това контролно НЕ се натрупват към точките за упражнение
- Условие за получаване на заверка – ако студентът има >10 т на упражнение и < 21т. трябва да събере поне 20 точки, ако студентът има >21 т. трябва да събере поне 10 т. от настоящия тест.
- Студентите НЯМАТ право да използват помощни материали.
- При опит за преписване или други опити за измама резултатите от това контролно се анулират.

ИМЕ: \_\_\_\_\_ ФАК. Но: \_\_\_\_\_ ГР. \_\_\_\_\_

## Типове от данни и бинарни операции

1. Да се пресметнат следните операции:

$(17+013)\ll 2 = ?$
---------------------

2. Да се пресметнат следните операции:

<pre>int x = 0x1537; int y; y = (((x&lt;&lt;3)&gt;&gt;5) 0252525)&amp;0x00ff; y = ? (десетично число)</pre>
---

3. Да се пресметнат следните операции:

<pre>char a, b = 0x20, c; a = ++b; --a; b--;- c = (a&amp;b)^016;</pre>
--

## Функции за вход и изход, логически операции, функции

4. Да се напише програмен фрагмент на функция, която приема като параметри 3 аргумента цели числа: a, b, c. Ако стойността на c е по-голяма от 150 функцията да върне като резултат логическо a ИЛИ b, ако стойността на c е по-малка от 25, функцията да върне остатъка от делението a/b. Ако стойността на c не отговаря и на двете условия да, функцията да върне сбора на a и b.
5. Да се напише програма, която прочита от клавиатурата 3 реални числа и намира най-голямото от тях
6. Да се напише програмен фрагмент, който проверява валидността на въведена променлива. За валидна стойност се приемат числата в интервала (5; 100]

## Цикли

7. Да се определи колко пъти ще се изпълни даденият цикъл:

```
int i, j=0;
for(i=0; j<=10; i++)
{
    if(i%3 == 2)
    {
        j++;
        i=0;
    }
    printf("Hello\n");
}
```

8. Да се определи колко пъти ще се изпълни следният цикъл:

```
int k = 7;
do
{
    k--;
}
while(k > 6);
```

9. Да се определи какво ще изведе на екрана следния програмен фрагмент:

```
int h = 15;
while( h > 0){
    printf("%x\n",h);
    h%=3;
}
```

## Масиви

10. Да се определи изхода от изпълнението на следната програма:

```
#include <string.h>
#include <stdio.h>
int main()
{
    int k;
    char g[30] = "Hello World!\n";
    for(k=0; k<30; k++)
        if(g[k] == '\0')
            g[k] = 0x00;
    printf("%s -> %d",g,strlen(g));
    return 0;
}
```

11. Да се определи изхода от изпълнението на следната програма:

```
#include <stdio.h>
void swap(int* a, int* b)
{
    int t = *a;
    *a = *b;
    *b = *t;
}
```

```

int main()
{
    int arr[3] = {1,3,2};
    int o,k;
    for(o=0; o<2; o++)
        for(k=o+1; k<3; k++)
            if(arr[o] > arr[k])
                swap(&arr[o],&arr[k]);
    for(k=0;k<3;k++)
        printf("%d ",arr[k]);
    return 0;
}

```

12. Да се напише програма, която прочита от стандартният вход число N ( $N < 100$ ), след което прочита N на брой цели положителни числа. Програмата да изведе средното аритметично на всичките N числа.

## Файлове

13. Да се напише програмен фрагмент, който отваря файл за четене и прочита от него едно едноцифрено число. Числото да се умножи по 2 и да се изведе на екрана.
14. Да се напише програмен фрагмент, който отваря бинарен файл за четене с опцията за добавяне.
15. Да се състави програма, в която се отваря текстови файл и в него се записват на отделен ред числата от 1 до 1000 в обратен ред.

## Многомерни масиви

16. Да се състави програмен фрагмент, в който се дефинира двумерен масив  $3 \times 3$ , а като начални стойности се задават числата от 1 до 9, като нарастването е по колони.
17. Да се състави програмен фрагмент, който извежда на екрана числата от главния диагонал на квадратна матрица  $5 \times 5$ .
18. Да се състави програмен фрагмент, който намира минималния елемент в двумерен масив и извежда колоната, където той се намира.

## Указатели

19. Да се определи изхода от изпълнението на следния програмен фрагмент:

```

char a[100] = "Programirane na C";
char *b;
for(b=a; *b!='m'; b++)
    printf("%c\n",*b);
    printf("%s\n",b);

```

20. Да се определи изхода от изпълнението на следния програмен фрагмент:

```

int a[5] = {3, 6, 2, 7, 8};
int k=0;
for(k =0; k<=4; k++)

```

```
if((*a+k) == k[a] && (a[k]&3 >2))
    printf("%d\n",k);
printf("PIK2");
```

21. Да се определи резултатът от изпълнението на следния програмен фрагмент:

```
float a[5]={3, 6, 2, 7, 8};
float *b = a;
printf("%d\n%d",sizeof(a),sizeof(b));
```

## Структури от данни

22. Да се дефинира структура, която описва размерите на цилиндър (радиус и големина)
23. Да се напише програмен фрагмент, който прочита от клавиатурата и записва в масив от структури информацията за 5 цилиндъра (Може да се използва дефинираната структура от 22.)
24. Да се напише програмен фрагмент, който заделя памет за 5 променливи от тип структура динамично и занулява заделената памет.

## Списъци

25. Да се дефинира структура, която описва елемент от едносвързан списък. Елементът да съхранява едно реално число.
26. Използвайки структурата от 25. да се създаде функция, която връща стойността на първия елемент, който има стойност по-малък от  $\pi$ , от свързания списък.
27. Да се направи съпоставка на едносвързаните списъци и статичните масиви, като се посочат най-малко три прилики или разлики.

## Машини на състоянията

28. Да се начертае блокова схема на преходите на машина на състоянието, която открива сложните изречения (изречения с една или повече запетайки между началото и края на изречението) в текстов файл.
29. Да се начертае блокова схема на преходите на машина на състоянието, която открива в текстови файл всички „думи“ от типа „ab...b“ (многоточията показват повтаряне на символа)
30. Да се напише програмен фрагмент, който реализира следната машина на състояния:

