11. Програмски језик Ц има наредбе које омогућавају промену тока управљања у неку другу тачку програма. Једна од наредби за промену тока извршења програма, омогућава:

- превремено завршавање циклуса (*for, while ili do-while*) скакањем на прву наредбу непосредно иза циклуса,
- прескакање следећих наредби унутар селекције *switch*
- очување структурности програма изван циклуса.

Помоћу које наредбе се постижу горе наведени искази:

1. continue
2. gotoxy
3. break
4. sizeof

1

13. Једна од наредби за промену тока извршења програма у програмском језику Ц, омогућава:

- у случају када се налази унутар наредби ***while*** и ***do-while,*** прелазак на поновно испитивање услова циклуса
- у случају наредбе ***for*** прелазак на извршавање израза 2 (тј. услова)
- у случају угњеждених циклуса, прескачу се само преостале наредбе најдубљег циклуса.
- наредба не нарушава структуираност програма ван циклуса
- ако се налази унутар наредбе **switch** која се налази унутар неког циклуса, скок се врши на крај тог циклуса уз наравно прескакање наредби које су биле унутар селекције **switch**.

1

Одредити за коју наредбуваже горе наведени искази:

1. continue
2. gotoxy
3. break
4. sizeof

56. Дат је Ц код, који након извршавања исцртава слику помоћу звездица.

```
int i,j,n=7;
for(i=1;i<=n/2;i++){
            for(j=1;j<=n/2-i+1;j++)  printf("  ");
            printf("*");
            for(j=1;j<=2*(i-1);j++) printf("  ");
      printf("\b*\n");
}
for(i=1;i<=n;i++)  printf("* ");
```

Анализирати дати код и одредити која слика ће бити исцртана након његовог извршавања.

57. Дата је декларација променљивих unsigned a,b,x и део кода у програмском језику Ц. Одредити шта се налази као резултат у променљивој x након извршења дате наредбе.

```
unsigned  a, b, x;
x=0;
while(a>=b){
      a-=b;
      x++;
}
```

1. Производ a и b
2. Збир a и b
3. Остатак приликом делења
4. Количник при дељењу a са b

2

58. Дата је декларација променљивих unsigned a,b, x,y, temp и део кода у програмском језику Ц. Одредити шта се налази као резултат у променљивој x и у након извршења датог кода.

```
unsigned  a, b, x, y, temp;
x=a*b;
while(b) temp=a%b, a=b, b=temp;
y=b;
x/=y;
```

1. X је производ a и b, а y је количник a са b
2. X је најмањи заједнички садржалац за a и b, а y највећи заједнички делилац за a и b
3. X је највећи заједнички делилац за a и b, а y најмањи заједнички садржалац за a и b
4. Без обзира на вредности променљивих, долази до грешке у последњој наредби кода
5. Долази до грешке јер петља понавља само прву наредбу услед изостанка витичастих заграда на телу петље

2

59. Дата је декларација променљивих pod,br и део кода у програмском језику Ц. Закључити шта представља вредност коју променљива br добије извршењем кода:

```
unsigned  pod, br ;
pod=128;
br=0;
while(pod!=0){
      if(pod & 0x1) br++;
      pod>>=0x1;
}
```

1. Број јединица у бинарном запису броја pod
2. Број нула у бинарном запису броја pod
3. Број цифара у бинарном запису броја pod
4. Број цифара у хексадецималном запису броја pod

2

60. Дат је део кода на програмском језику Ц, који контролише унос целобројне променљиве n. Одредити вредности које променљива n може добити.

```
do{
        printf("Unesite N:\nN = ");
        scanf("%d",&n);
        if(n & 1) printf("Greska.\n");
}while(n & 1);
```

1. Омогућава унос непарног природног броја
2. Омогућава унос само позитивног природног броја
3. Омогућава унос само негативног природног броја
4. Омогућава унос парног природног броја
5. Омогућава унос само непарног позитивног природног броја

2

76. Наведени су искази који се односе на дефиницију while циклуса. Који од ових исказа су **тачни**:

1. While циклус се извршава све док је услов логичка неистина (једнак нули),
2. While циклус се користи када се зна колико ће се пута циклус извршавати,
3. У while циклусу се увек прво проверава да ли је услов логичка истина, те ако јесте наредба се извршава,
4. Код while циклуса се може десити да се тело циклуса не изврши ниједном (на почетку услов није задовољен).

1

77. Наведени су искази који се односе на дефиницију do while циклуса. Који од ових исказа су **тачни**:

1. Користи се када се не зна колико ће се пута циклус понављати.
2. Прво се извршава тело циклуса, а затим израчунава вредност логичког израза. Ако се добије логичка неистина циклус се поновно извршава.
3. Циклус се завршава када услов добија вредност логичке истине.
4. Циклус се извршава барем једном.

1