ПРОГРАМИРАЊЕ – ПРОГРАМСКИ ЈЕЗИК С

У следећим задацима заокружите број испред траженог одговора

1. Заокружити број испред кључне речи којом се при кодирању у програмском језику Ц, у наредби вишестуког гранања обележавају вредности за које се улази у поједине гране:

   1. switch
   2. break
   3. return
   4. case

1

2. У програмском језику Ц декларисане су променљиве и дат је део кода програма:

```
FILE *fp;
char str[150];
fp=fopen("primer.txt", "r");
fgets(str, 80, fp);
```

Одредити шта је последица извршавања датог кода. Заокружити број испред очекиваног одговора:

1. Учитава 80 карактера из датотеке и смешта у стринг str.
2. Учитава максимално 150 карактера из датотеке и смешта у стринг str
3. Учитава стринг из датотеке све док се не учита знак за нови ред или 80 карактера
4. Учитава стринг из датотеке све док се не учита знак за нови ред или 150 карактера

1

3. У програму написаном у програмском језику Ц декларисана је променљива **pod** типа **int**. Употребом функције **fprintf(...)** уписати декларисан податак у стандардну излазну датотеку.

Заокружити број испред исправно написане наредбе:

1. `fprintf(pod);`
2. `fprintf("%d", pod);`
3. `fprintf("%d", pod, stdin);`
4. `fprintf(stdout, "%d", pod);`

1

4. У програмском језику Ц декларисани су структурни типови података ***Tacka3D*** (који дефинише тачку у простору) и ***Lopta*** (одређена центром и полупречником):

```
typedef struct
{
  float x, y, z;
}Tacka3D;
```

```
typedef struct
{
  Tacka3D centar;
  float R;
}Lopta;
```

Заокружити број испред исправно написане наредбе декларације и иницијализације променљиве ***x*** типа ***Lopta***, тако да јој центар буде у тачки O(2,2,2), а полупречник 10цм:

```
1. Lopta x={10, {2, 2, 2}};
2. Lopta x={2, 2, 2, 10};
3. Lopta x={2, 2, 2}, {10};
4. Lopta x={{2, 2, 2}, 10};
```

1

5. У програму написаном у програмском језику Ц декларисана је променљива fp која представља показивач на бинарну датотеку и променљива **podatak** у коју ће се уписати прочитани подаци из дефинисане бинарне датотеке.

Заокружити редни број испред наредбе која омогућава учитавање три бајта са тренутне позиције из бинарне датотеке:

1. `fread(podatak, 24, 1, fp);`
2. `fread(&podatak, 24, 0, fp);`
3. `fread(&podatak, 3, 1, fp);`
4. `fscanf(&podatak, 3, 1, fp);`
5. `fscanf(fp, 3, &podatak);`

1

6. У програму написаном у програмском језику Ц декларисана је променљива fp која представља показивач на бинарну датотеку и променљива **podatak** чија вредност ће се уписати у дефинисану бинарну датотеку.

Заокружити редни број испред наредбе која омогућава упис три бајта у бинарну датотеку:

1. `fprintf(&podatak, 3, 1, fp);`
2. `fprintf(fp, 3, &podatak);`
3. `fwrite(podatak, 24, 1, fp);`
4. `fwrite(&podatak, 24, 0, fp);`
5. `fwrite(&podatak, 3, 1, fp);`

1

7. У програмском језику Ц дата је наредба декларације, а затим и наредба форматираног излаза:

```
float x = 5.56;
printf(" x = %f\tx = %g\n", x, x);
```

Након извршења ових наредби на екрану ће се приказати вредности променљивих у задатом формату. Заокружити број испред тачног одговора:

1. x = 5.560000e+000 x= 0
2. x = 5.560000 x = 5.560000e+000
3. x = 5.560000 x = 5.56
4. x = 5.56 x = 5.560000e+000

2

8. У програмском језику Ц декларисана је целобројна променљива и додељена јој је вредност логичког израза:

```
int x;
x = 1==10>5;
```

Имајући у виду приоритет оператора, одредити вредност променљиве x после извршења ове наредбе. Заокружити број испред траженог одговора:

1. променљива добија вредност логичке неистине, тј. x = 0
2. променљива добија вредност логичке истине, тј. x = 1
3. вредност логичког израза се не може доделити целобројној променљивој
4. променљива добија вредност логичке истине, тј. било који број различит од 0

2

9. У програмском језику Ц декларисане су две целобројне променљиве. Променљива **a** добија вредност уносом са тастатуре. Затим се вредност логичког израза додељује променљивој **x**:

```
int x, a;
scanf("%d", &a);
x = 10!=5 || a<2;
```

Имајући у виду приоритет оператора, одредити вредност променљиве x после извршења ове наредбе. Заокружити број испред траженог одговора:

1. уколико се заградама не одредити редослед извршавања операција у овом изразу, долази до грешке, тј. „пуцања“ програма
2. без обзира на вредност која се унесе у променљиву **a**, вредност израза је увек „тачно“, тј. **x = 1**
3. без обзира на вредност која се унесе у променљиву **a**, вредност израза је увек „нетачно“, тј. **x = 0**
4. вредност израза зависи од променљиве **a** и не може се једнозначно одредити уколико није позната вредност уписана у променљиву **a**

2

10. Наредбом у Ц језику треба проверити да ли је број паран или непаран. Проценити која од датих наредби врши ову проверу и заокружити број испред тачно написане наредбе.

1. `(broj % 2 == 1) ? printf("PARAN!!") : printf("NEPARAN!!");`
2. `(broj % 2) ? printf("PARAN!!") : printf("NEPARAN!!");`
3. `(broj % 2 == 0) ? printf("PARAN!!") : printf("NEPARAN!!");`
4. `(broj & 1) ? printf("PARAN!!") : printf("NEPARAN!!");`
5. `(broj & 0x1 == 0) ? printf("PARAN!!") : printf("NEPARAN!!");`
6. `(broj & 1 == 1) ? printf("PARAN!!") : printf("NEPARAN!!");`

2

11. Код дат у тексту задатка треба реализовати помоћу једне if наредбе. Заокружити број испред понуђеног тачног одговора:

```
if(x>1)
{
      if(x<6)
            y=4;
}
```

1. `if(x>1 && x<6 ) y=4;`
2. `if(x>1 || x<6 ) y=4;`
3. `if(x<1 || x>6 ) y=4;`
4. `if(!(x<=1 || x>=6)) y=4;`

2

12. Дат је део кода на програмском језику Ц:

```
for(j=0; j<n; j++)
      if(a[j]>0)s+=a[j];
      else break;
```

Свака for петља може се написати коришћењем while и do-while наредбе. Заокружити број испред понуђеног кода који је еквивлентан коду датом у тексту задатка:

1.
```
j=0;
while(j<n && a[j]>0) s+=a[j++];
```
2.
```
j=0;
while(j<n && a[j++]>0) s+=a[j];
```
3.
```
j=0;
while(j<n || a[j]>0) s+=a[j++];
```
4.
```
j=0;
while(j<n && a[j]<=0) s+=a[j++];
```

2

13. Дата је декларација променљивих unsigned a, b и део кода у програмском језику Ц. Одредити шта се налази као резултат у променљивој x и у након извршења датог кода. Заокружити број испред траженог одговора:

```
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 а и б, а у је количник а са б
2. x је најмањи заједнички садржалац за а и б, а у највећи заједнички делилац за а и б
3. x је највећи заједнички делилац за а и б, а у најмањи заједнички садржалац за а и б
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

14. Дата је декларација променљивих pod, br и део кода у програмском језику Ц:

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

Закључити шта представља вредност коју променљива br добије извршењем кода. Заокружити број испред траженог одговора:

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

15. Дат је део кода на програмском језику Ц, који контролише унос целобројне променљиве n. Одредити вредности које променљива n може добити. Заокружити број испред траженог одговора:

```
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

16. Дата је декларација променљивих и део програмског кода:

```
int i, temp, n = 11;
int x[30]={ -3, -1, -2, -2, 1, 4, 3, 1, 5, -8, 5};
temp=x[0];
i=0;
while(i<n-1) x[i++]=x[i+1];
x[n-1]=temp;
```

Просудити на основу наредби које ће бити извршене у **while** циклусу како ће изгледати трансформисан низ **x од n елемената**. Заокружити број испред очекиваног одговора:

1. `x[ ] ={ 5, -3, -1, -2, -2, 1, 4, 3, 1, 5, -8 }`
2. `x[ ] ={ -1, -2, -2, 1, 4, 3, 1, 5, -8, 5, -3 }`
3. `x[ ] ={ -2,  0, -1, -1, 2, 5, 4, 2, 6, -7, 6 }`
4. `x[ ] ={ -1, -2, -2, 1, 4, 3, 1, 5, -8, 5 }`

2

17. У програмском језику Ц је дата декларација променљивих, а касније у коду извршен позив функције на следећи начин:

```
int k, i;
char lista[10][50], ime[50];
if( Formiraj(lista[i], ime, k) == NULL) { ... }
```

На основу позива, проценити каквог је облика прототип функције и заокружити број испред тачно написаног прототипа:

1. `void *Formiraj(char s1, char s2, int x);`
2. `char Formiraj(char *s1, char *s2, int x);`
3. `int *Formiraj(char s1[], char s2[], int x);`
4. `int Formiraj(char s1[], char s2[], int x);`
5. `char *Formiraj(char s1, char s2, int x);`

2

18. У програмском језику Ц је дата декларација променљивих, а касније у коду извршен позив функције на следећи начин:

```
int x, y, i, j;
float **mat, *vek, z;
mat[i] = Formiraj(x, 0.5);
```

На основу позива, проценити каквог је облика прототип функције и заокружити број испред тачно написаног прототипа:

1. `float Formiraj(int n, float m);`
2. `void *Formiraj(int n, int m);`
3. `float *Formiraj(float n, int m);`
4. `float *Formiraj(int n, float m);`
5. `float **Formiraj(int n, float m);`

2

19. Дат је код рекурзивне функције написан у програмском језику Ц:

```
void prikaz(int k, int n){
      printf("%d\t",k);
      if(k<n) prikaz(k+1,n);
      printf("%d\t",k);
}
```

Проценити шта ће се десити ако се функција позове наредбом: `prikaz(4,10);`

2

Заокружити број испред тачног одговора:

1. 4 5 6 7 8 9 10
2. 4 5 6 7 8 9 10 9 8 7 6 5 4
3. 4 5 6 7 8 9 10 10 9 8 7 6 5 4
4. 10 9 8 7 6 5 4

20. У програмском језику Ц дат је прототип функције **funkcija()** и декларисане су променљиве у функцији **main().** У понуђеним одговорима дати су позиви функције за декларисане променљиве.

```
void funkcija(int *x, int *y, int **p);
void main(){
      int a=5, b=7, c=15, *poc;
      poc = &c;
}
```

2

Заокружити редни број испред исправно записаног позива декларисане функције:

1. `funkcija(a, b, &poc);`
2. `funkcija(&a, &b, &poc);`
3. `funkcija(&a, &b, poc);`
4. `c = funkcija(&a, &b, &poc);`

21. У програмском језику С декларисани су структурни типови података ***Tacka*** (одређена координатама)***, Poligon*** (одређен бројем и координатама темена) и ***Piramida*** (одређена типом основе – троугао, четвороугао... и висином). Потом је декларисана и променљива типа ****Piramida***:

```
typedef struct
{
  float x, y;
}Tacka;
```

```
typedef struct
{
  int brojTemena;
  Tacka temena[10];
}Poligon;
```

```
typedef struct
{
  Poligon osnova;
  float visina;
}Piramida;
```

2

```
Piramida *p;
```

Заокружити број испред наредбе којом се број темена основе пирамиде на коју показује декларисани показивач ****p***, поставља на 6:

1. `p.osnova.brojTemena=6;`
2. `p.osnova->brojTemena=6;`
3. `p->osnova.brojTemena=6;`
4. `p->osnova[brojTemena]=6;`
5. `p->osnova->brojTemena=6;`

У следећим задацима заокружите бројеве испред тражених одговора

22. Заокружити бројеве испред **ТАЧНИХ** исказа који се односе на дефиницију while циклуса: 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**
23. Наведени су искази који се односе на дефиницију **do-while** циклуса. Заокружити бројеве испред **ТАЧНИХ** исказа: 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**
24. Заокружити бројеве испред понуђених тврдњи које представљају тачне наставке изјаве која се односе на повратну вредност функције **fopen**: **При покушају да датотеку отворимо за читање, функција fopen...** 1. ако датотека не постоји, изазива грешку која доводи до пуцања програма 2. ако датотека не постоји, креира празну датотеку, поставља се на њен почетак и враћа показивач на ту датотеку 3. враћа NULL показивач ако датотека не постоји 4. ако датотека постоји, враћа показивач на ту датотеку	**1**
25. Дате су наредбе декларације променљивих (са и без иницијализације вредности) написане на програмском језику Ц. Заокружити бројеве испред исправно написаних наредби декларације променљивих: 1. `int a=b=c=5;` 2. `int a=5, b=5, c=5;` 3. `char zn="a";` 4. `long a; b=5; c;` 5. `int a=0xf2;` 6. `char zn='\b';`	**1,5**

26. Декларисане су следеће променљиве:
`float x, z;`
`const float y;`
Заокружити бројеве испред НЕИСПРАВНО написаних наредби доделе вредности променљивама:
1. x %= y;
2. x =+ 5;
3. x += y + 5;
4. x =/ y + 5;
5. y = x + z;
6. x = z = y + 5;
	1,5

27. Дата је наредба декларације **int a, b;**
Имајући у виду дату декларацију, заокружити бројеве испред НЕИСПРАВНО написаних наредби форматираног уноса података:

1. `scanf("%d%f", &a, &b);`
2. `scanf("%d%*d", &a, &b);`
3. `scanf("%d%d", &a, &b);`
4. `scanf("%d%d", a, b);`
5. `scanf("%d%*d", &a);`
6. `scanf("%5d%5d", &a, &b);`

1,5

28. Дате су наредбе декларације и иницијализације једнодомензионалног низа целих бројева у програмском језику Ц. Заокружити бројеве испред исправно написаних наредби декларације и иницијализације једнодимензионалног низа:

1. `int a[10]={1,2,3};`
2. `int a[5]={-3, -2, -1, 0, 1, 2, 3};`
3. `int a[]={10,20,30,40,50};`
4. `int[5] a={1, 2, 3, 4, 5};`
5. `int a={10,20,30,40,50};`
6. `int a[5]={'1', '2', '3', '4', '5'};`

1,5

29. Термин „адресна аритметика“ се односи на извођење аритметичких операција над показивачима. Анализирати дате исказе који дефинишу дозвољене аритметичке операције над показивачима. Заокружити бројеве испред тачних исказа:

1. Додела вредности једног показивача другом.
2. Додавање рационалног податка на вредност показивача и одузимање рационалног податка од вредности показивача.
3. Одузимање и упоређивање два показивача.
4. Идентификатор низа је показивач на почетак низа и може му се мењати вредност.
5. Упоређивање показивача са **NULL**.
6. Сабирањем два показивача, добија се нови показивач.

1,5

30. Наредбама програмског језика Ц дата је декларација једне симболичке константе и једне константне променљиве:

```
#define k 50 …
int m=100; …
```

Заокружити бројеве испред исправно написаних наредби декларације дводимензионалног низа целих бројева (матрице):

1. `int a [ k ][ k ];`
2. `int b [ k ][ m ];`
3. `int c [ k ][ 10 ];`
4. `int x [100 ][ 50];`
5. `int y [10, 10];`
6. `int z [ m ][ 10 ];`

1,5

31. У следећем задатку заокружити бројеве испред тражених одговора.
Дата је наредба у Ц језику, која температуру у Целзијусима ***tempc*** претвара у температуру у Фаренхајтима ***tempf***. Подаци tempc и tempf су реални бројеви обичне тачности. Проценити који изрази дају тачно решење.

1. temf = (9 / 5) * temc + 32;
2. temf = 9 / 5 * temc + 32;
3. temf = 9 * temc / 5 + 32;
4. temf = 32 + 9 * temc / 5;

2

32. Дата је if-else наредба:

```
if(a==3 || a==5)  p++;
else p--;
```

Заокружити бројеве испред понуђених switch наредби које су еквивалентне датој if-else наредби:

1.
```
switch(a){
            case 3: p++; break;
            case 5: p++; break;
            default: p--;
}
```
2.
```
switch(a){
            case 3: case 5: p++; break;
            p--;
}
```
3.
```
switch(a){
            case 3: case 5: p++; break;
            default: p--;
}
```
4.
```
switch(a){
            case 3: case 5: p++;
            default: p--;
}
```

2

33. У програму на програмском језику Ц извршена је следећа декларација, а касније и резервација меморијског простора за низ реалних бројева обичне талности дужине n:

```
float *B;
int n;
B=(float*)calloc(n,sizeof(float));
```

Заокружити бројеве испред исправно написаних наредби за **ПРИКАЗ** i-тог елемента низа B:

1. `printf("%f", B[i]);`
2. `printf("%f", &B[i]);`
3. `printf("%f", B+i);`
4. `printf("%p", *(B+i));`
5. `printf("%f", *(B+i));`

2

34. У програму на програмском језику Ц извршена је следећа декларација, а касније и резервација меморијског простора за низ реалних бројева обичне талности дужине n:

```
float *B;
int n;
B=(float*)calloc(n,sizeof(float));
```

Заокружити бројеве испред исправно написаних наредби за **УНОС** i-тог елемента низа B:

1. `scanf("%f", B[i]);`
2. `scanf("%f", B+i);`
3. `scanf("%p", B+i);`
4. `scanf("%f", &B[i]);`
5. `scanf("%f", *(B+i));`

2

35. Дати су прототипови функција написани у програмском језику Ц. Заокружити бројеве испред исправно написаних прототипова функција:

1. `float* pp1(int a, int b, int c);`
2. `int pp2(int a[][10], int n);`
3. `int pp3(int a[], n; float b);`
4. `void pp4(int *a, int n);`
5. `int pp5(int a[][], int n);`
6. `int pp6(int a[], int n);`
7. `int pp7(int a, b, c);`
8. `float[ ] pp8(float a[ ], int n);`

2

36. Заокружити бројеве испред понуђених тврдњи које представљају тачне наставке изјаве која се односе на повратну вредност функције **fopen**:

При покушају да датотеку отворимо за <u>писање</u>, функција fopen...

1. ако датотека не постоји, креира празну датотеку, поставља се на њен почетак и враћа показивач на ту датотеку
2. враћа NULL показивач ако датотека не постоји
3. ако датотека постоји, излази упозорење да ће њен садржај бити уништен при отварању
4. ако датотека не постоји, изазива грешку која доводи до пуцања програма
5. ако датотека постоји, уништава њен садржај без упозорења

2

37. У програмском језику Ц је декларисана низовна променљива:

```
int niz[10];
```

Заокружити бројеве испред исправно написаних наредби читања низа целих бројева дужине 10 из бинарног фајла на који показује показивач ***in**:

1. `fread(niz, sizeof (int), 10, in);`
2. `fread(&niz, sizeof (int), 10, in);`
3. `fread(&niz, sizeof niz, 1, in);`
4. `fread(niz, sizeof niz, 1, in);`
5. `fread(niz, sizeof (niz), 1, *in);`
6. `fread(niz, sizeof (int)*10, in);`

2

38. У програмском језику C декларисан је структурни тип података ***Ucenik***, а затим и променљива типа ***Ucenik***:

```
typedef struct
{
    char ime[50];
    int razred;
    int ocene[10];
}Ucenik; …
int i; Ucenik x;
```

Заокруживањем обележити исправне начине приступа пољима структурне променљиве ***x***:

1. `x.ocene[i]`
2. `*x.razred`
3. `x->ime`
4. `x[i].ocene`
5. `x.ime`

2

39. У програмском језику Ц декларисан је структурни тип података ***Putovanje***, а затим и променљива типа ****Putovanje***:

```
typedef struct
{
    char start[50], cilj[50];
    int kilometraza;
}Putovanje; …
Putovanje *p;
```

Заокруживањем обележити исправне начине приступа пољима структурне променљиве:

1. `*p->kilometraza`
2. `(*p).kilometraza`
3. `&p->kilometraza`
4. `p->start`
5. `*(p).start`

2

40. Дат је прототип функције написан у програмском језику Ц:

```
void Saberi(int n, int *a, int *b);
```

У main функцији дате су следеће декларације променљивих:

```
int x[50][50], y[50], m, j, i;
```

Заокружити бројеве испред исправно написаних позива декларисане функције:

1. `Saberi(m, y[i], y[i+1]);`
2. `Saberi(y[i], x[i], x[i+1]);`
3. `Saberi(m, y, x[i][j]);`
4. `Saberi(y, x[i], x[i+1]);`
5. `Saberi(10, y, x[0]);`
6. `Saberi(x[i][j], x[i], x[j]);`

3

41. Дат је прототип функције написан у програмском језику Ц:

```
void Umetni(char *a, char k);
```

У main функцији дате су следеће декларације променљивих:

```
char s1[20], *s2, s3;
```

Заокружити бројеве испред исправно написаних позива декларисане функције:

1. `Umetni(s2, s1[i]);`
2. `Umetni(s2, s1);`
3. `Umetni(s2, 'A');`
4. `Umetni(s1, s3);`
5. `Umetni(*s2, s3);`
6. `Umetni(s3, &s1);`

3

Допуните следеће реченице и табеле

42. Дата су наредба декларације, а затим и наредба форматираног уноса вредности у променљиве, написана на програмском језику Ц:

```
int x, y;
scanf("%3i%3i", &x, &y);
```

Следи тастатурни унос у облику: **12345 12345**
За сваку променљиву одредити и на одговарајућу линију уписати, коју ће вредност променљива имати по извршењу наредби:

1. променљива x добија вредност **x =** _____
2. променљива y добија вредност **y =** _____

2

43. Дати је декларација променљивих **`int a=3, b=15;`**

Израчунати вредност коју ће променљиве имати по извршењу следеће наредбе:

```
b %= ++ a;
```

a = _____

b = _____

2

44. Одредити вредности које ће променљиве x и y имати по извршењу следећег кода:

```
int x=10;
int y=20;
if(x>50)
      x-=10;
      y+=10;
```

Уписати добијене вредности на предвиђене линије:

x = _____ y = _____

2

45. У програмском језуку Ц декларисане су две целобројне променљиве:

```
int x=0, izbor;
```

За дате вредности промељиве ***izbor***, одреди вредност промељиве *x* по извршењу следеће наредбе вишестуког гранања и уписати их на предвиђене линије:

```
switch(izbor)
{
case 1: x += 1;
case 2: x += 2; break;
case 3: x += 3;
default: x = 100;
case 4: x += 4;
case 5: x += 5;
}
```

1. за izbor=3, x=_____
2. за izbor=10, x=_____
3. за izbor=4, x=_____
4. за izbor=2, x=_____

2

46. Наредбама програмског језика Ц декларисана је правоугаона матрица и три целобројне променљиве:

```
int mat[10][20];  int x, N, M;
```

где **N** представља број врста, а **M** број колона правоугаоне матрице **mat**.

Допунити изразима који недостају кôд петље која има задатак да дуплира све елементе **последње колоне** матрице:

```
for(x = 0; x < __________; x++)

    mat[__________][__________] *= 2;
```

2

47. Дата су следеће декларације: `int p[200], i, n, k;`

А затим и део кода који треба да из низа **p** дужине **n,** сажимањем **ИЗБАЦИ** елеменат низа са позиције k, а затим ажурира нову дужину низа.

Имајући у виду дату иницијализацију петље, у предвиђена поља унеси одговарајуће елементе **преписивањем израза** из листе понуђених израза (подразумевати да су све потребне променљиве иницијализоване):

```
for(i=k; i ________;________)

     ________ = ________;

n--;
```

1. p [i + 1]
2. p [i - 1]
3. p [i]
4. p [k]
5. i ++
6. i - -
7. < n
8. < n-1

2

48. Наредбама програмског језика Ц декларисана је правоугаона матрица и три целобројне променљиве:

```
int mat[10][20];  int k, N, M;
```

где **N** представља број врста, а **M** број колона правоугаоне матрице **mat**.

Допунити изразима који недостају кôд петље која има задатак да дуплира све елементе **прве врсте** матрице:

```
for(k=0; k< __________; k++)

    mat[__________][__________]*=2;
```

2

49. Дати су изрази формирани коришћењем математичких оператора. Водећи рачуна о типовима података, одредити вредности датих израза и уписати их линију у продужетку. Ако израз изазива грешку, уместо вредности, написати **error**:

1. 10 / 4. = _____
2. 10. / 5 = _____
3. -10 % 3 = _____
4. 10. % 5 = _____
5. 10 % (-3) = _____
6. (100/3) % 6 = _____

3

50. Дате су следеће декларације: `int p[200], i, n, pom;`

А затим и део кода који треба да врши циклично померање елемената низа **p** дужине **n,** за једно место **удесно**. У коду недостају неки од елемената.

Имајући у виду дату иницијализацију петље, у предвиђена поља унеси одговарајуће елементе **преписивањем израза** из листе понуђених израза (подразумевати да су све потребне променљиве иницијализоване):

```
pom = ________;

for(i=n-2; i ________; ________)

      ________ = ________;

________ = pom;
```

1. p [0]
2. p [n-1]
3. p [n]
4. p [i + 1]
5. p [i - 1]
6. p [i]
7. i ++
8. i - -
9. >=0
10. >0

3

51. Дата су следеће декларације: `int p[200], i, n, k, x;`

А затим и део кода који треба да у низ **p** дужине **n** УБАЦИ (инсертује) елемент x на позицију k, а затим ажурира нову дужину низа.

Имајући у виду дату иницијализацију петље, у предвиђена поља унеси одговарајуће елементе **преписивањем израза** из листе понуђених израза (подразумевати да су све потребне променљиве иницијализоване):

```
for(i=n; i ________;________)

      ________ = ________;

________ = x;

n++;
```

1. p [i + 1]
2. p [i - 1]
3. p [i]
4. p [k]
5. i ++
6. i - -
7. >= k
8. > k

3

52. У програмском језику Ц, дате су следеће декларације: `int A[50], i, n;`

Потребно је формирти вектор са следећим вредностима:

i=0	i=1	i=2	i=3	i=4	i=5	…	i=n-1
1	2	4	7	11	16	…	???

Допунити програмски код којим се формира овај вектор:

```
A[0]=1;

for(i = __________; i __________; i++)

      __________ = __________;
```

3

53. У програмском језику Ц, декларисане су и иницијализоване променљиве:

```
int x=40, y=50, z=60, *p1, *p2;
```

Одреди које ће вредности имати променљиве x, y и z после извршења следећег кода и упиши на одговарајућу линију:

```
p1 = &x;
p2 = p1;
y = (*p2)+20;
z = *p2;
```

x = _____; y = _____; z = _____;

3

54. Дат је део кода написан на програмском језику Ц:

```
int a[7]={10,25,30,15,40,77,45}, *pa, x, y;
pa=a+4;
x=--(*pa)+5;
y=*(--pa)+5;
```

Анализирати код и одредити вредности променљивих **x** и **y**, као и показивача **pa,** по извршењу све три извршне наредбе датог кода:

x = _____

y = _____

pa = a + _____

3

55. Дат је део кода написан на програмском језику Ц:

```
int a[7]={81,12,35,97,40,52,17}, *pa, x, y;
pa=a+3;
x = *(pa-2)+1;
y = (*pa-2)+1;
```

Анализирати код и одредити вредности променљивих **x** и **y**, као и показивача **pa,** по извршењу све три извршне наредбе датог кода:

x = _____

y = _____

pa = a + _____

3

56. Дата је дефиниција функције:

```
void Transformisi(float *x, float *y, float z)
{
    z++;
    *x=*x+z;
    (*y)++;
}
```

У главном програму су декларисане променљиве и извршен је позив функције:

```
float a=10, b=10, c=10;

Transformisi(&a, &b, c);
```

Одредити које вредности имају променљиве a, b и c по изласку из функције и уписати их на одговарајућу линију:

a = _____

b = _____

c = _____

3

57. На програмском језику Ц декларисан је и иницијализован стринг и два показивача:

```
char s1[]="Short Message Service", *s2, *s3;
```

Одредити и на предвиђену линију уписати садржај означених стрингова по извршењу следећих наредби:

```
s2=strchr(s1,'M');
s3=strrchr(s2,'S');
strncpy(s1+1,s2,1);
strcpy(s1+2,s3);
```

s1 = _____

s2 = _____

s3 = _____

3

У следећим задацима уредите и повежите појмове према захтеву

58. Са леве страни дати су допунски параметри у функцији printf , а са десне стране значење тих параметара у програмском језику Ц. На линију испред значења унети број којим је означен одговарајући допунски параметар:

1. **(#)** _____ означава да ће се поравнавање вршити уз леву ивицу поља ширине **n** знакова, допунски знакови размака додају се иза, а не испред податка
2. **(0)** _____ означава да се испред позитивног броја мора исписати знак плус
3. **(-)** _____ нула код нумеричких података означава да ће се приликом равнања уз десну ивицу број допуњавати нулама, а не знаковима размак
4. **(+)** _____ исписује се децимална тачка у конверзији рационалних бројева који немају разломљени део.

2

59. Декларисана је реална променљива `float w=123.456;`:

Са леве стране су дати различити прикази вредности променљиве добијени коришћењем наредби форматираног излаза које су приказане са десне стране. Поред сваке наредбе, на предвиђену линију уписати рени број приказа добијеног изршавањем те наредбе:

1. 123.456000 _____ `printf("%g", w);`
2. 1.234560e+002 _____ `printf("%f", w);`
3. 123.456 _____ `printf("%.2f", w);`
4. 123.46 _____ `printf("%e", w);`

2

60. Са леве стране дати су математички изрази, а са десне запис израза на програмском језику Ц. На линију испред записа израза, унети број којим је означен одговарајући израз:

1. $y = \frac{\sqrt{x}+10}{a+|b|}$
2. $y = \frac{\sqrt{x+10}}{a}+|b|$
3. $y = \frac{\sqrt{x+10}}{a+|b|}$
4. $y = \sqrt{x}+\frac{10}{a}+|b|$

_____ y = sqrt(x+10) / (a+fabs(b))

_____ y = sqrt(x)+10 / a+fabs(b)

_____ y = sqrt(x+10) / a+fabs(b)

_____ y = (sqrt(x)+10) / (a+fabs(b))

2

61. Дат је код на програмском језику Ц:

```
switch(c){
            case 'A': case 'a': printf("Pravougaonik ");
            case 'B': case 'b': printf("Trougao "); break;
            case 'C': case 'c': printf("Krug ");
            default: printf("Duz "); break;
}
```

Са десне стране су дате вредности променљиве с (скретница), а са леве стране резулат извршења кода за дату вредност скретнице. На линију испред вредности скретнице унети редни под којим је наведен одговарајући екрански приказ:

1. Krug Duz _____ 'b'
2. Pravougaonik Trougao Krug Duz _____ 'K'
3. Krug _____ 'A'
4. Trougao _____ 'c'
5. Pravougaonik Trougao
6. Duz

2

62. Са леве стране су набројани различити типови променљивих, а са десне су дате декларације променљивих у програмском језику Ц. На линију испред декларације унети редни број под којим је наведен одговарајући тип променљиве:

1. Једнодимензионални низ показивача на целе бројеве _____ `int *a;`
2. Вектор целих бројева _____ `int a[100];`
3. Показивач на цео број _____ `int a*[100];`
4. Цео број _____ `int *a[100];`
5. Грешка у декларацији

2

63. У програмском језику Ц, декларисан је показивач на цео број и функцијом **calloc** додељен му је простор за смештај низа од **n** целих бројева:

```
int *a, n;
scanf("%d", &n);
a=(int*)calloc(n, sizeof(int));
```

У левој колони дати су изрази, а у десној опис њиховог значења. На линију испред сваког од израза унеси број којим је означено одговарајуће објашњење:

_____ `&a[0];`	1. вредност елемента на последњој позицији у низу
_____ `*(a+n-1);`	2. адреса четвртог елемента у низу
_____ `a+4;`	3. адреса почетног елемента низа
_____ `*a;`	4. вредност елемента на предзадњој позицији низа
	5. вредност елемента на почетној позицији у низу
	6. адреса петог елемента у низу

2

64. Наредбама програмског језика Ц декларисано је дводиманзионално поље реалних бројева (матрица) и три целобројне променљиве:

```
float mat[10][10];  int i, j, n;
```

где променљива **n** представља димензију квадратне матрице **mat**.

Са леве стане су дате ознаке елемената матрице, а са десне њихово тумачење. На линију испред сваке ознаке унети редни број одговарајућег тумачења:

_____ `mat[j][n-1]`	1. елеменат у j-тој врсти и последњој колони
_____ `mat[j]`	2. и-та врста матрице
_____ `mat[0][j]`	3. j-та врста матрице
_____ `mat[i]`	4. j-та колона матрице
	5. елеменат у првој врсти и j-тој колони
	6. грешка у нотацији

2

65. Са леве стране су наведене наредбе позиционирања у датотеци, а са десне описи ефеката датих наредби. На линију поред наредбе уписати редни број под којим је наведен опис ефекта наредбе:

ftell(dat)	_____	1. позиционирање на почетак датотеке
fseek(dat, 0, SEEK_END)	_____	2. позиционирање на крај датотеке
fseek(dat, 0, SEEK_SET)	_____	3. одређује позицију у датотеци у виду броја бајтова од почетка датотеке
rewind(dat)	_____	4. ништа од понуђеног

2

66. Са леве стране дате су врсте конверзије, а са десне типови података који се користе у функцији за приказ података printf у програмском језику Ц. На линију испред типа података унеси број којим је означена одговарајућа конверзија:

1. %d
2. %i
3. %s
4. %ld
5. %f
6. %e
7. %hd
8. %u

_____ short
_____ signed int (u dekadnom obliku)
_____ long
_____ unsigned
_____ signed int (dekadni, heksadekadni ili oktalni oblik)

2,5

67. Са леве стране наведене су функције за читање и упис у текст датотеку, а са десне стране опис функције. На линију испред описа функције унети редни број под којим је наведена одговарајућа функција:

1. fscanf _____ учитавање карактера из датотеке
2. fgets _____ учитавање реда из датотеке
3. fputs _____ форматирани упис података у датотеку
4. fprintf _____ упис стринга у датотеку
5. fgetc _____ форматирано учитавање података из датотеке

2,5

68. Са леве стране су набројани неки од прелазних знакова тј. escape секвенце, а са десне стране дати су њихови описи. На линију испред описа упишите број под којим је наведена одговарајућа escape секвенца:

1. '\n' _____ враћање на почетак реда (carrage return)
2. '\t' _____ системски звучник (bell)
3. '\r' _____ прелаз у нови ред (new line)
4. '\b' _____ није escape секвенца
5. '\h' _____ хоризонтални табулатор (horizontal tab)
6. '\a' _____ враћање једну курсорску позицију назад (backspace)

3

69. Са десне стране наведене су неке од функција библиотеке ctype.h, а са леве су дати њихови описи. Испред назива сваке од наведених функција, уписати редни број под којим је дат одговарајући опис:

1. Да ли је с штампајући знак (укључујући и размак)? _____ isspace(c)
2. Да ли је с велико слово? _____ isdigit(c)
3. Да ли је с знак интерпункције? _____ isalpha(c)
4. Да ли је с управљачки знак? _____ isupper(c)
5. Да ли је с децимална цифра? _____ iscntrl(c)
6. Да ли је с знак бели знак? _____ isprint(c)
7. Да ли је с слово?
8. Да ли је с хекса-децимална цифра?

3

70. На програмском језику Ц декларисане су променљиве:

```
char s1[]="Iwnt2CmyM8sagain", *sn;
```

Са леве стране написани су изрази доделе вредности стрингу **sn**, а са десне стране понуђене су вредности стринга **sn**. На линију написати редни број под којим је наведена вредност стринга **sn** која се добија извшењем одговарајућег израза:

_____ `sn=strrchr(s1,'a')-1;`	1. NULL
_____ `sn=strchr(s1,'a')+1;`	2. “in”
_____ `sn=strstr(s1,"my");`	3. “ain”
_____ `sn=strstr(s1,"T2");`	4. “gain”
	5. “sagain”
	6. “myM8sagain”

4