

Präsenzaufgabe: Konstruktoren und Destruktoren

Geben Sie die Ausgabe des folgenden Programms an.

```
1  /*** Aufgabe.h ***/
2  /* Klasse: Alpha */
3  class Alpha {
4  private:
5      char x;
6  public:
7      Alpha();
8      Alpha(char y);
9      ~Alpha();
10 };
11
12 /* Klasse: Bravo */
13 class Bravo {
14 private:
15     char x;
16 public:
17     Bravo();
18     Bravo(char y);
19     Bravo(char x, char y);
20     ~Bravo();
21 };
22
23 /* Klasse: Charlie */
24 class Charlie : public Bravo, public Alpha {
25     char x;
26 public:
27     Charlie();
28     Charlie(char y);
29     ~Charlie();
30     void f(char c);
31 };
32 /*** Ende Aufgabe.h ***/

```



```
1  /*** Aufgabe.cpp ***/
2  #include <iostream>
3  #include "Aufgabe.h"
4  using namespace std;
5
6  /*** Klasse: Alpha ***/
7  Alpha::Alpha() : Alpha('A') {
8      cout << "Alpha\t" << x << endl;
9  }
10
11 Alpha::Alpha(char y) : x('a') {
12     char t = x;
13     x = y;
14     cout << "Alpha\t" << x << endl;
15     x = t;

```

```

16 }
17
18 Alpha::~Alpha() {
19     cout << "-Alpha\t" << x << endl;
20 }
21
22 /***** Klasse: Bravo *****/
23 Bravo::Bravo() : Bravo('B') {
24     x++;
25     cout << "Bravo\t" << x << endl;
26 }
27
28 Bravo::Bravo(char y) : x(y) {
29     cout << "Bravo\t" << x << endl;
30 }
31
32 Bravo::Bravo(char x, char y) : x(x) {
33     cout << "Bravo\t" << x << endl;
34     cout << "Bravo\t" << y << endl;
35 }
36
37 Bravo::~Bravo() {
38     cout << "-Bravo\t" << x << endl;
39 }
40
41 /***** Klasse: Charlie *****/
42 Charlie::Charlie() : Charlie('C') {
43     x = 'c';
44     cout << "Charlie\t" << x << endl;
45 }
46
47 Charlie::Charlie(char y) : x(y) {
48     cout << "Charlie\t" << x << endl;
49 }
50
51 Charlie::~Charlie() {
52     cout << "-Charlie\t" << x << endl;
53 }
54
55 void Charlie::f(char c) {
56     Alpha b(c);
57     Bravo f(c+1, c+2);
58 }
59 /***** main-Funktion *****/
60 int main() {
61     Charlie c;
62     c.f('X');
63     return 0;
64 }
65 /*** Ende Aufgabe.cpp ***/

```

Präsenzaufgabe: Zeiger

Gegeben sei das folgende C++-Programm:

```
1  #include <iostream>
2  #define macro { int u; int v; char* w; char* y; }
3  using namespace std;
4
5  int main() {
6      char a[] = { 97, 98, 99, 100, 101, 0 };
7      char c = 'A', d = 's';
8      char* pa;
9      char* p;
10     //0
11
12     pa = &a[3];
13     *(pa-2) = *pa-3;
14     //1
15
16     p = &a[4];
17     *(a+4) -= 5;
18     (*p)++;
19     //2
20
21
22     d -= *a;
23     a[0] = d+*(&c);
24     //3
25
26     struct macro s,
27     *ps = &s;
28     (*ps).u = 11;
29     ps->v = s.u+5;
30     ps->w = a+2;
31     s.y = s.w+1;
32     *(*ps).w += s.u;
33     *(*ps).y += (&s)->v;
34     //4
35 }
36
37 //End
```

Wir interessieren uns für die Werte der Variablen an den mit Zeitpunkt 1, Zeitpunkt 2 usw. gekennzeichneten Zeilen. Vervollständigen Sie die Tabelle.

Zeitpunkt	a[0]	a[1]	a[2]	a[3]	a[4]	a[5]	c	d	s.u	s.v	s.w	s.y
0	97	98	99	100	101	0	-	-	-	-	-	-
.	.		.	.		.	-	-	-	-	-	-
.									.			